INPUT 64

DAS ELEKTRONISCHE MAGAZIN 8/85

Infos · News · Programme · Unterhaltung · Tips

Wettbewerbssieger:
DiscMon 1541

Diesmal im Raum:
Geo mit Nico

8 Textfenster gleichzeitig:
Window-Technik mit dem C64

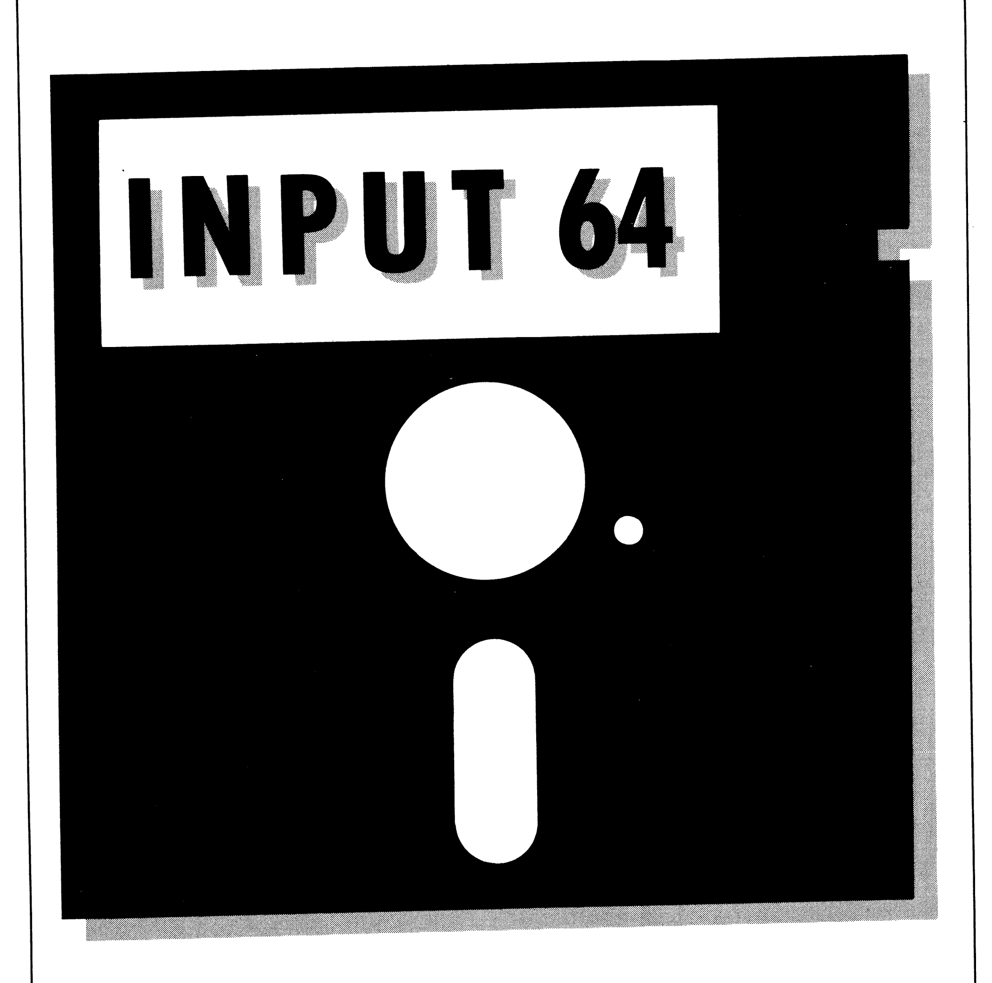
Abgehört:
Klaviaturen im Härtetest

Spielwiese:
- **Paulchens Schiebung**
- **Klicker**
- **Colorcode**

und außerdem natürlich:
64er Tips
Hilfsprogramme
Soundkurs

Dokumentation und Bedienungshinweise

INPUT - ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1


INPUT 64





8/85

Liebe 64er-Besitzer(in),

wir sind ratlos. Fördert INPUT 64 die Kriminalität? Oder haben sich die Raubkopierer in Ihrem illegalen Tun auf Autoeinbrüche verlegt? Es geht um folgende Geschichte: Kürzlich stellte einer unserer Redakteure in einem der nobleren Stadtteile Hannovers sein (nicht mehr ganz so nobles) Auto ab. Als er zurückkam, war eine Seitenscheibe eingeschlagen und seine Tasche samt Scheckheft, Fahrzeugpapieren und Personalausweis entwendet. Das Sperren der Schecks erwies sich allerdings als überflüssig: Die Tasche wurde bereits von einem Polizeibeamten gefunden, und auch der Inhalt war weitgehend komplett. Es fehlte nur eins: die neuste Ausgabe von INPUT 64.

Was sollen wir davon halten?
Wie auch immer:

viel Spaß mit INPUT 64

Hinweise zur Bedienung

1.) Entfernen Sie - bei ausgeschaltetem Rechner - eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.

2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD"INPUT*",8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.

3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise ("Bitte Band zurückspulen" und so weiter ...).

4.) Merken Sie sich fünf Kommandos:

- (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.

- (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres

funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert. Von Diskette wird automatisch neu geladen.)

- (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

- (CTRL) und (s) steht für SAVEn und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwenderprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisungen - Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programmanfang möglich - entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

- (CTRL) und (b) ermöglicht einen Bildschirmausdruck - natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Voraussetzung ist Geräteadresse 4 oder 6.

6.) Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist.

Auf der Rückseite der Verpackung finden Sie das Inhaltsverzeichnis der Kassette/Diskette.

Auf einen Blick: INPUT 64-Betriebssystembefehle

Inhaltsverzeichnis aufrufen	CTRL und i
Hilfsseite aufrufen	CTRL und h
Programm abspeichern	CTRL und s
Bildschirmausdruck	CTRL und b
Titelbild abkürzen	CTRL und q

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner "abstürzen". Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

Leser fragen...

Eigene Vokabeln für DICTIONARY

" ... Betrifft: INPUT 64 Ausgabe 4/85 - DICTIONARY. Das Programm mit GOTO 1800 zu starten hat bei mir nicht zum Erfolg geführt. Meine Version ist: 1.) Vokabeln laden. 2.) Bei "Filename" RETURN drücken. 3.) Geräteadresse 3 eingeben (und RETURN). Es erscheint das Menue, dann F2 (neue Vokabeln eingeben) wählen: Freie Bytes 31114.

(Th. Klüsener, Berlin)

Die von Herrn Klüsener beschriebene Möglichkeit, das Programm DICTIONARY mit einer eigenen Vokabeldatei zu betreiben, ist in der Tat eleganter als die von uns beschriebene. Die Geräteadresse 3 steht übrigens für "Eingabe vom Bildschirm", diesen Trick kann man auch ausnutzen, um sich alle eingegebenen Vokabeln zeigen zu lassen: "Vokabeln speichern" wählen und als Geräteadresse "3" eingeben!

(d.Red.)

Nico und "SAVEN"

" ... MATHE MIT NICO wollte ich aus dem Magazin auf meinen eigenen Datenträger kopieren. Leider gelang mir dies in der Ausgabe 6/85 nicht, obwohl dies bei Ausgabe 5/85 problemlos möglich war!

(tel. Anfragen)

Alle Kurse und Serien innerhalb von INPUT 64 sind grundsätzlich nicht zum Speichern auf den eigenen Datenträger freigegeben. Wir gehen aber davon aus, daß wir mit dem Angebot an Programmen, die für den Eigenbedarf abgespeichert werden können (Anwenderprogramme, Spiele, Hilfsprogramme), ein sehr offenes Software-System anbieten. Hinzu kommt, daß die nicht zum Abspeichern freigegebenen Programme in der Regel nicht ohne das jeweilige INPUT-Betriebssystem lauffähig sind.

(d.Red.)

SuperTape und Dateibehandlung

" ... gibt es anscheinend Probleme beim Wiedereinlesen sequentieller Dateien vom Band mit SuperTape DII. Die Zeile mit dem Schreibbefehl lautet:
FOR I= 1 TO N: PRINT#1, D$(N);: NEXT
Will ich diese Daten durch die Schleife
FOR I=1 TO N: INPUT#1, E$(N): NEXT
wieder einlesen, erhalte ich immer die Fehlermeldung "string too long error"!

(H. Genz, Hamburg)

Der hier auftretende Fehler hat nichts mit SuperTape zu tun, fehlerverursachend ist das Semikolon in der ersten Zeile. Der Befehl "PRINT#1,

D$(N);" schreibt nämlich alle Daten ohne abschließendes RETURN (CHR$(13)) nach jedem Datum hintereinander aufs Band. Der Befehl INPUT liest aber solange Zeichen ein, bis er auf ein RETURN stößt. Da dieses Zeichen nicht gefunden wird, kommt es nach dem Einlesen des 255. Zeichens zum "string too long error", denn mehr Zeichen nimmt der INPUT-Befehl nicht an. Besagter Fehler wäre also auch beim Schreiben/Lesen auf Diskette oder auf Kassette im Commodore-Format aufgetreten.

"Syntax error" bei PRINT AT und INKEY

" ... Beim Starten der PRINT AT/INKEY Routine am BASIC-Anfang erhalte ich fast immer die Meldung "syntax error". Außerdem ist mir aufgefallen, daß die Farbe der eingegebenen Buchstaben oder Zahlen gelegentlich undefiniert wechselt ... "

(tel. Anfrage)

Das erste Problem taucht nur beim ersten Start der Routine auf. Da ein eventuell hinter der Routine liegendes BASIC-Programm mit gestartet werden soll, wird ein RUN ausgegeben. Ist nun noch kein sinnvoller Code vorhanden (sprich: kein BASIC-Programm), kommt es zur Meldung "syntax error". Nicht verwirren lassen, ein NEW bereinigt den Speicher, und es kann programmiert bzw. nachgeladen werden.

Das zweite Problem taucht bei bestimmten Rechnerserien auf. Abhilfe schafft immer, vor dem Aufruf des INKEY-Befehls das entsprechende Eingabefeld durch den PRINT AT-Befehl mit einem Leerstring zu löschen.

(d.Red)

HAUSHALTSBUCHFÜHRUNG

" ... habe ich die HAUSHALTSBUCHFÜHRUNG (Ausgabe 6/85) auf Kassette abgespeichert und neu geladen. Wenn ich nach dem Eingeben der Daten erneut "saven" will, kommt immer die Meldung 'Das INPUT 64-Betriebssystem befindet sich im Rechner. Zur Zeit kann von diesem Programm ausgehend nicht gespeichert werden. Dieses Programm bitte erst mit CTRL s auf Ihren eigenen Datenträger abspeichern.' !"

(tel. Anfrage)

Setzen Sie vor die Anweisungen in den Zeilen 5070 und 6230 ein REM. Dort wird anhand einer signifikanten Speicherstellenbelegung abgefragt, ob das INPUT-Betriebssystem noch im Rechner ist. Es hat sich allerdings herausgestellt, daß bei manchen Rechnern eben diese Adresse auch im Einschaltzustand denselben Inhalt hat wie bei Anwesenheit des INPUT-Betriebssystems.

(d.Red.)

Paulchen's Schiebung

Kennen Sie eigentlich noch diese kleinen Puzzles, bei denen man innerhalb eines Rahmens kleine Teile mit aufgedruckten Buchstaben hin- und herschieben mußte, bis man eine bestimmte Wortkombination zusammen hatte?

Bei Paulchen's Schiebung ist das so ähnlich. Mit Joystick oder Tastatur müssen Sie die einzelnen Teilstücke so verschieben, daß sich wieder ein vollständiges Bild ergibt.

Auf dem Bild ist der Kopf eines Panthers zweimal abgebildet. Mit der Leertaste beziehungsweise dem Feuerknopf des Joysticks wird das Bild gemischt. Durch einen zweiten Tastendruck beenden Sie das Mischen.

Falls Sie keinen Joystick haben: Taste 'A' = Hoch
Taste 'Z' = Runter
Taste '.' = Rechts
Taste ',' = Links

Mathe mit Nico

Nico wird diesmal Ihre Kenntnisse der Volumen- und Oberflächenberechnung von Hohlkörpern überprüfen. Selbstverständlich in seiner geduldigen und netten Art.

Ihnen steht auch diesmal wieder eine Rechenseite zur Verfügung. Sie erreichen diese Seite, wenn Sie "R" und RETURN drücken:

Sie können die folgenden Rechenoperationen benutzen:

Rechenoperation	Beispiel
+ = Plus	17 + 4
- = Minus	12 - 9.9
* = Mal	35 * 34.8
/ = Geteilt durch	87 / 14.9

Jede Zeile muß mit einem RETURN abgeschlossen werden. Sie können auch mehrere Berechnungen in einer Zeile ausführen lassen, jedoch sind Klammern nicht erlaubt. Bis zur Bestätigung durch die RETURN-Taste können Sie die Zeile frei editieren.

Wenn Sie die Rechenergebnisse auf die Ausgabenseite übergeben möchten, geben Sie bitte ein "L" und RETURN ein.

Sie können sich aber auch den Rechenweg wieder ausführlich von Nico erklären lassen.

Wir wünschen Ihnen wieder viel Spaß mit "Mathe mit Nico".

REISEKOSTEN

Urlaubszeit - Reisezeit. Damit die Kasse stimmt, will die Fahrt gut geplant sein.

Sie finden im Magazin ein Programm, mit dem Sie ermitteln können:

- Wie weit die Entfernung zwischen Abfahrts- und Zielort ist,
- wie hoch die Benzinkosten für die jeweilige Strecke sind,
- mit welchen Gesamtkosten Sie für Ihr Fahrzeug rechnen müssen
- und wie die Deutsche Bundesbahn im Kostenvergleich abschneidet.

Besondere Vergünstigungen der DB wie rosarote Aktionen können vom Programm natürlich nicht berücksichtigt werden. Übrigens bieten auch Bahnunternehmen anderer Länder besondere Vergünstigungen an, die bei den Auskunftsstellen der DB teilweise erfragt werden können.

Innerhalb der BRD und der DDR stehen Ihnen 47 Orte zur Auswahl zur Verfügung. Hinzu kommen noch 16 Städte des Europäischen Auslandes.

Sie können Abweichungen von Abfahrt- oder Zielort zusätzlich angeben, falls Sie von einem Ort in der Nähe losfahren oder zu einem Ort in der Nähe des Zielortes wollen.

Im Bereich des Europäischen Auslandes sind die Kosten für eine Fahrt mit der Bahn nur für direkte Fahrten zu den Zielorten genau. Der Fahrpreis je 100 km hängt vom jeweiligen Land ab und bewegt sich zwischen 7,50 DM und 29.00 DM

Land	Preis/je 100km
Belgien	13,40 DM
Dänemark	13,30 DM
Italien	7,60 DM
Niederlande	22,50 DM
Norwegen	26,60 DM
Österreich	20,20 DM
Schweden	19,40 DM
Schweiz	27,00 DM
Tschecheslowakei	20,50 DM

Zur Berechnung der Gesamtkosten für Kraftfahrzeuge wurde die ADAC-Betriebskostentabelle zugrunde gelegt. Die Ausgaben je Kilometer können in Abhängigkeit vom Hubraum ermittelt werden. Für die Hubraumstaffelung wurde der Mittelwert aller Modelle bis zu der jeweiligen Hubraumklasse ermittelt:

Hubraum bis ccm	Pfg./km
800	28,6
1000	33,8
1200	35,9
1500	39,8
1800	45,7
2000	53,4
über 2000	66,5
über 3000	94,9

Die Bahnkilometer wurden mit den PKW-Reisekilometern gleichgesetzt. Von 100-1000 km wurde ein Bahnkilometer mit 18,5 Pfg berechnet, bis 2000 mit 17,5 Pfg.

Es wurde ein möglichst kostengerechter Vergleich angestrebt. Obwohl in extremen Fällen Abweichungen auftreten können, ist ein objektiver Kostenvergleichvergleich in allen Fällen gewährleistet.

Programmtechnisch wurde die Berechnung der Entfernung zwischen ausgewählten Orten über eine Entfernungstabelle ermittelt, wie Sie sie aus Taschenkalendern kennen. Der Vorteil liegt in der relativ einfachen Programmierbarkeit, der Nachteil im zunehmenden Aufwand an Zahleneingabe, wenn man einen weiteren Ort hinzunehmen möchte.

Bei Reisen in Europäische Länder wurde folgende Lösung gewählt:
Jeder ausländischen Stadt wurde eine Stadt in der BRD als Grenz- beziehungsweise Durchfahrtsort zugeordnet. Diese Zuordnung schränkt die Anwendung des Programms zwar etwas ein, wurde aber möglichst sinnvoll festgelegt, wobei auch die Autobahngebühren in anderen europäischen Ländern mit bedacht wurden. So ist es beispielsweise oft günstiger, über Aachen nach Paris zu fahren, weil dadurch die Kosten für Autobahnbenutzung in Frankreich eingespart werden können.

In diesem Zusammenhang wäre ein Programm interessant, das in einem Verkehrsnetz die günstigste Entfernung zwischen Abfahrts- und Zielort findet; quasi selbst mit dem Finger über die Landkarte fährt. Ein solches Programm müßte dabei erheblich schneller sein als der eigene Finger, da es sonst den Aufwand nicht lohnt

Sollte Ihnen hierzu eine Lösungsmöglichkeit einfallen, können Sie uns gerne etwas dazu schreiben. Stichwort: NETZWEG.

Die Bedienung des Programms:

Die Hauptmenuepunkte:

- Verzeichnis der Städte
- Berechnung über Städte
- Berechnung ueber Kilometer

werden mit Hilfe der Cursortasten (auf/ab) ausgewählt. Mit RETURN wird die gewünschte Auswahl angesprochen

Im "Verzeichnis der Städte" können sie mit der SPACE-Taste blättern. Mit 'm' gelangen Sie zurück ins Menue.

Bei "Berechnung über Städte" wählen Sie zuerst den Abfahrtsort - ebenfalls über die Cursor-Taste (auf/ab). Auch hier können Sie das Gewünschte mit RETURN übernehmen.

Für die Auswahl der Zielorte gilt Entsprechendes, nur das hier zusätzlich Städte des Europäischen Auslandes angewählt werden können.

Nachdem die Entfernung zwischen den ausgewählten Orten angezeigt wurde, können Sie nach Wunsch Plus- oder Minuskilomter angeben. Dies ist sinnvoll, wenn Sie von einem Ort in der Nähe einer Großstadt aus fahren wollen.

Nach der Entfernungsfestlegung erfragt das Programm die Grundangaben für die Fahrzeugkosten.
Dies sind:

- Benzinverbrauch in Liter j 100 km
- Benzinpreis in DM je Liter
- Hubraumgröße des benutzten Fahrzeuges in ccm

Benzinverbrauch und Benzinpreis werden direkt eingegeben. Der Hubraum kann entweder über Cursortaste (auf/ab) ausgesucht oder aber direkt eingegeben werden.

Geben Sie Hubraum 0000 ccm ein, können Sie selbst eine Kilometerpauschale festlegen.

"Berechnung über Kilometer" erwartet von Ihnen die Angabe der Entfernung in Kilometern. Anschließend werden Sie zu den Grundangaben zur Fahrzeugkosten-Berechnung geführt.

Der Lohn der Mühe sind zwei Bildschirmseiten, auf denen die Reisekosten-Berechnungen dargestellt werden:
- Fahrzeugkosten
- Vergleich Kraftfahrzeug - Bundesbahn

Innerhalb von Input 64 können Sie eine Hardcopy der Bildschirme auf Ihren Drucker ausgeben.

Folgende Städte:

Aachen	Emden	Kassel	Obersdorf
Augsburg	Erfurt	Kiel	Passau
Baden-Baden	Essen	Koblenz	Regensburg
Basel	Flensburg	Köln	Saarbrücken
Berlin	Frankfurt/M.	Konstanz	Salzburg
Bielefeld	Freiburg i.B.	Leipzig	Stuttgart
Bonn	Garmisch-Part.	Lübeck	Trier
Braunschweig	Hamburg	Magdeburg	Ulm
Bremen	Hannover	Mannheim	Wiesbaden
Dortmund	Heidelberg	München	Würzburg
Dresden	Hof	Münster	Wuppertal
Düsseldorf	Karlsruhe	Nürnberg	

Europäische Städte im Ausland

Zielort	über:
Amsterdam	Aachen
Ankara	München
Barcelona	Basel
Bruxelles	Aachen
Budapest	München
Helsinki	Flensburg
Istanbul	München
Luxemburg	Aachen
Madrid	Basel
Oslo	Flensburg
Paris	Aachen
Rom	München
Salzburg	München
Stockholm	Flensburg
Wien	München
Zürich	Basel

Programm-Beschreibung

Zeile	1-38	Titelbild, Einlesen DATAs
Zeile	39-61	Menue-Auswahl
Zeile	62-90	Städteverzeichnis
Zeile	91-98	Kilometer Eingabe
Zeile	99-146	Gesamtkosten je km Eingabe
Zeile	147-235	Reisekosten Ausgabe
Zeile	236-298	Auswahl Abfahrts- und Zielort
Zeile	299-312	Fehlermeldung

```
1 poke56,7*16+15:poke55,255
2 x2$="qwertyuioplkjhgfdsazxcvbnm./-"
3 x1$="0987654321."
4 y1$="       Reisekosten-Berechnung        "
5 bl$="                                    "
6 ln$="―――――――――――――――――――――――――――――――――――――"
7 b$(1)="0800":b$(2)="1000":b$(3)="1200":b$(4)="1500":b$(5)="1800"
8 b$(6)="2000":b$(7)="2000":b$(8)="3000":b$(9)="0000"
9 dimal$(63),k1(50,50),ak(16),zl$(16),iz(16),pz(16)
10 za=47:poke53281,7:poke53280,12
11 ik=2096:pr=2691:pz(0)=0.
12 :
13 print""+chr$(14)+chr$(8);y1$
14 syspr,2,8,"(von  Peter Hoelterhoff)"
15 syspr,7,14,"Guten Tag! "
16 syspr,11,10,"Bitte etwas Geduld!!"
17 a1$="Verzeichnis der Staedte"
18 a2$="Berechnung ueber Staedte"
19 a3$="Berechnung ueber Kilometer"
20 :
21 forz1=1to63
22 :readal$(z1)
23 nextz1
24 :
25 forz2=1to46
26 :za=za-1
27 :forz3=1toza
28 :readk1(z2,z3)
29 :nextz3
30 nextz2
31 :
32 foru1=1to16
33 :readzl$(u1),iz(u1),pz(u1)
34 nextu1
35 :
36 foru2=1to16
37 :readak(u2)
38 nextu2
39 print""y1$"":z9=0
40 poke53281,3:syspr,5,13,"Menue-Auswahl":poke646,0:gosub57:gosub54
41 syspr,22,3,"Auswahl ueber Cursortasten (auf/ab)"
42 poke198,0:ff=1
43 :
44 poke646,3:sysik,21,20,1,"",in$,f:poke646,0
45 gosub57
46 iff=1thenff=ff+1
47 iff=-1thenff=ff-1
48 iff=0then53
49 ifff=0thenff=3
50 ifff=4thenff=1
51 onffgosub54,55,56
52 goto44
53 onffgosub62,236,91
54 syspr,10,8,""a1$"":return
55 syspr,13,8,""a2$"":return
56 syspr,16,8,""a3$"":return
57 syspr,10,8,a1$
58 syspr,13,8,a2$
59 syspr,16,8,a3$
60 return
61 goto43
62 poke53281,15:poke53280,11
63 print""y1$
64 syspr,3,10,"Staedte  Verzeichnis"
65 print" Beachten Sie die Hinweise im Beiheft!"
66 fors=1to12:syspr,6+s,1,a1$(s):nexts
67 fors=13to24:syspr,s-6,21,a1$(s):nexts
68 syspr,22,1,"<space> weiter        m Menue":poke198,0
69 getx$:ifx$=""then69
70 ifx$="m"then39
71 ifx$=" "then73
72 goto69
73 gosub90:poke646,1
74 fors=25to36:syspr,s-18,1,a1$(s):nexts
75 fors=37to47:syspr,s-30,21,a1$(s):nexts
76 syspr,22,1,"<space> weiter        m Menue":poke198,0
77 getx$:ifx$=""then77
78 ifx$="m"then39
79 ifx$=" "then82
80 goto77
81 :
82 gosub90:poke646,5
83 fors=48to58:syspr,s-41,1,a1$(s):nexts
84 fors=59to63:syspr,s-52,21,a1$(s):nexts
85 syspr,22,1,"<space> zurueck       m Menue   "
86 getx$:ifx$=""then86
87 ifx$="m"then39
88 ifx$=" "thengosub90:goto66
89 goto86
90 fori=1to12:syspr,6+i,1,bl$:nexti:return
91 print""y1$:print"  Geben Sie Ihre Reise-Entfernung in"
92 print"  Kilometer ein!"
93 syspr,10,10,"Strecke:          km":syspr,10,19,"       ":poke198,0
94 sysik,10,19,7,x1$,km$
95 km=val(km$):z9=1:ifkm>10000orkm<1then93
96 syspr,21,12,"<space> weiter"
97 poke198,0
98 getx$:ifasc(x$+chr$(0))<>32then98
99 print""y1$:poke53281,12:poke53280,11
100 syspr,3,3,"Wieviel Liter Benzin braucht ihr"
101 syspr,4,3,"Fahrzeug im Schnitt?"
102 syspr,5,3,"(Beachten Sie, ob Sie ueberwiegend)"
103 syspr,7,3,"Autobahn,Landstr.,Stadt fahren"
104 syspr,10,3,"Verbrauch in Liter/100 km:":syspr,10,31,"      "
105 sysik,10,31,5,x1$,v1$:v1=val(v1$)
106 ifv1>35orv1<1then104
107 syspr,13,3,"Ein Liter kostet (DM) :"
108 syspr,13,31,"       ":sysik,13,31,6,x1$,kt$:kt=val(kt$)
109 ifkt>5.then108
110 v2=v1/100*km:ko=v2*kt:kb=v1*kt/100:kb=int((kb+0.005)*100)/100
111 print""y1$
112 syspr,3,1,"Wie gross ist der Hubraum des PKW?"
113 syspr,22,3,"Auswahl ueber Cursor oder Eingabe"
114 syspr,10,12,"Hubraum in ccm"
115 z5=1
116 ifz5<=6thensyspr,11,12,"bis   "
117 ifz5>6thensyspr,11,12,"ueber"
118 syspr,11,19,b$(z5):poke198,0
119 sysik,11,19,4,x1$,q$,f
120 iff=0then126
121 iff=1thenz5=z5+1
122 iff=-1thenz5=z5-1
123 ifz5<1thenz5=9
124 ifz5>9thenz5=1
125 goto116
126 ifval(q$)=0then136
127 hr=val(q$):ifhr>9999then114
128 ifhr<801thenhb=.286:goto147
129 ifhr<1001thenhb=.338:goto147
130 ifhr<1201thenhb=.359:goto147
```

```
131 ifhr<1501thenhb=.398:goto147
132 ifhr<1801thenhb=.457:goto147
133 ifhr<2001thenhb=.534:goto147
134 ifhr<3000thenhb=.665:goto147
135 hb=.949:goto147
136 print""y1$
137 syspr,3,1,"Geben Sie Ihre Gesamtkosten pro Kilo-"
138 syspr,4,1,"meter ein.(in DM z.B. 0.42)"
139 syspr,6,1,"Die Gesamtkosten beinhalten:"
140 syspr,8,1,"Treibstoff-,Werkstatt-,Versicherungs-"
141 syspr,9,1,"und aehnliche Kosten!"
142 syspr,22,1," z zurueck zur Hubraum-Eingabe"
143 syspr,15,1,"Kilometer-Pauschale:            ":poke198,0:syspr,15,31,"DM"
144 syspr,15,24,"       "
145 sysik,15,24,6,x1$+"z",hb$:hb=val(hb$):ifhb>5.orhb<.01then143
146 ifhb$="z"then111
147 print""y1$:poke53281,15
148 syspr,3,2,"PKW-Reisekosten fuer einfache Fahrt"
149 ifz9=1then153
150 ifz9=2then154
151 ifz9=3then155
152 ifz9=5then156
153 syspr,6,1,"Preis fuer Entfernung":goto157
154 syspr,6,1,"";a1$(z2);" - ";a1$(z3);:goto157
155 syspr,7,1,"(Abweichung um ";k1;"km eingegeben)"
156 syspr,6,1,"";a1$(z2);" - ";a1$(q3);:goto157
157 xx=km:gosub233:syspr,6,30-px,km:syspr,6,37,"km"
158 syspr,8,1,"Treibstoff-Kosten DM/km"
159 syspr,8,31,kb:syspr,8,37,"DM"
160 syspr,10,1,"Verbrauch in Liter    ":rv=int((v2+0.005)*100)/100
161 xx=rv:gosub233:syspr,10,30-px,xx
162 syspr,12,0,ln$
163 syspr,14,1,"Treibstoff-Kosten  ca. "
164 rv=int((ko+0.005)*100)/100
165 xx=rv:gosub233:syspr,14,30-px,rv:syspr,14,37,"DM"
166 syspr,16,0,ln$
167 kg=km*hb
168 syspr,18,1,"Gesamtaufwand ca. "
169 xx=int(kg):gosub233:syspr,18,30-px,xx:syspr,18,37,"DM"
170 syspr,21,0,ln$
171 syspr,22,17,"<space>"
172 getx$:ifx$=""then172
173 ifx$=" "then175
174 goto172
175 ifkm<100or(km>2000andz9=1)then198
176 ifkm<=1000thend1=0.185:goto178
177 ifkm>1000thend1=0.17:goto178
178 dk=km*d1+pz(q5):da=((dk/100)*50)+dk
179 print""y1$
180 syspr,4,17," PKW "
181 syspr,6,1,"Treibstoff-Kosten        ca. "
182 xx=int(ko):gosub233:syspr,6,30-px,xx:syspr,6,37,"DM"
183 syspr,8,1,"Gesamt-Kosten            ca. "
184 xx=int(kg):gosub233:syspr,8,30-px,xx:syspr,8,37,"DM"
185 syspr,10,0,ln$
186 syspr,12,17," DB  "
187 syspr,14,1,"1.Klasse                 ca. "
188 xx=int(da):gosub233:syspr,14,30-px,xx:syspr,14,37,"DM"
189 syspr,16,1,"2.Klasse                 ca. "
190 xx=int(dk):gosub233:syspr,16,30-px,xx:syspr,16,37,"DM"
191 syspr,18,0,ln$
192 syspr,20,4,"Angaben nur fuer einfache Fahrt"
193 syspr,22,1," <space> zurueck  w neue Berechnung"
194 getx$:ifx$=""then194
195 ifx$="w"thenak=0:goto39
196 ifasc(x$)=32then147
197 goto194
198 print""y1$""
199 syspr,2,2,"Vergleich mit der Bundesbahn erfolgt"
200 syspr,3,2,"nur ab 100 bis 2000 Kilometer."
201 ifkm>2000thengoto221
202 syspr,5,5,"Bereich"," Kl.2","  Kl.1"
203 syspr,6,6,"km",,"  DM","   DM"
204 syspr,8,6,"1-5",," 1.80","  2.80"
205 syspr,9,6,"6-10"," 2.40","  3.60"
206 syspr,10,5,"11-15"," 3.20","  4.80"
207 syspr,11,5,"16-20"," 3.80","  5.80"
208 syspr,12,5,"21-30"," 5.00","  7.60"
209 syspr,13,5,"31-40"," 7.00"," 10.60"
210 syspr,14,5,"41-50"," 9.00"," 13.60"
211 syspr,15,5,"51-60","11.00"," 17.00"
212 syspr,16,5,"61-70","13.00"," 20.00"
213 syspr,17,5,"71-80","14.00"," 21.00"
214 syspr,18,5,"81-90","16.00"," 24.00"
215 syspr,19,5,"91-100","18.00"," 27.00":poke198,0
216 syspr,22,8,"<space> zurueck  w weiter"
217 getx$:ifx$=""then217
218 ifx$="w"thenak=0:goto39
219 ifasc(x$)=32then147
220 goto217
221 syspr,5,1,"Ihre Reise-Entfernung liegt ueber"
222 syspr,6,1,"2000 Kilometer."
223 syspr,8,7,"Folgende Preise als Beispiel"
224 syspr,9,14,"(je 100 km)"
225 syspr,11,1,"Belgien:    13.40 DM  Oesterr: 20.20 DM"
226 syspr,12,1,"Daenemark: 13.30 DM  Schweiz: 27.-- DM"
227 syspr,13,1,"Italien:    7.60 DM  Schweden 19.40 DM"
228 syspr,14,1,"Niederl:    22.50 DM  Tschech. 20.50 DM"
229 syspr,15,1,"Norwegen:  26.60 DM"
230 syspr,19,1,"Bitte informieren Sie sich ueber den"
231 syspr,20,1,"Fahrpreis bei der Deutschen Bundesbahn"
232 goto216
233 ifxx=0then235
234 px=int(log(int(xx)+.1)/log(10))
235 return
236 print""y1$:z9=2:poke53281,1
237 syspr,8,5,"Abfahrts-Ort:"
238 z2=21
239 syspr,8,20,"                "
240 syspr,8,20,a1$(z2):poke198,0
241 sysik,8,20,15,"",z2$,f
242 iff=1thenz2=z2+1
243 iff=-1thenz2=z2-1
244 iff=0thenz3=1:goto248
245 ifz2>47thenz2=1
246 ifz2<1thenz2=47
247 goto239
248 qk=0:z3=1
249 syspr,10,5,"Zielort    :"
250 syspr,10,20,"                "
251 syspr,10,20,a1$(z3):poke198,0
252 sysik,10,20,15,"",z2$,f
253 iff=1thenz3=z3+1
254 iff=-1thenz3=z3-1
255 iff=0thengoto259
256 ifz3=<1thenz3=63
```

Fortsetzung Seite 44

Kurs komplett

Jetzt als Sampler:
Die Serie BITS & BYTES IM VIDEO-CHIP

Alle Folgen des Kurses aus den Ausgaben eins bis fünf sind ab sofort gesammelt auf Kassette und Diskette erhältlich. Eine grundlegende Einführung in die Programmierung des Video-Chips, mit Exkursen in die Binärarithmetik, Programmiertips und so weiter.

Überarbeitet und um einen Teil zur Multicolor-Grafik erweitert.

Erscheinungsdatum: 15.9.85

Jetzt bestellen und 5 Mark sparen! Für Bestellungen bis zum 1.9.85 gibt es den Sampler zum Subscriptionspreis:

Kassette 12.80 DM (mit SuperTape-Lader und Sicherheitskopie auf der Rückseite)

Diskette 19.80 DM

jeweils inclusive Porto und Verpackung.

Bestellungen bitte direkt an den

Verlag Heinz Heise GmbH
- Vertrieb
Bissendorfer Str.8

3 Hannover 61

(Verrechnungsscheck oder quittierten Einzahlungsbeleg beilegen, Bestellschein auf der Rückseite)

Für Bestellungen, die nach dem 1.9.85 eingehen, kostet die Kassette 17.80 DM, die Diskette 24.80 DM.

Bitte abtrennen und im Briefumschlag versenden!

Kurs Komplett-Bestellschein

Ja, ich möchte 5 Mark sparen und bestelle den gesammelten Kurs BITS & BYTES IM VIDEO-CHIP

○ auf Kassette zum Subscriptionspreis von 12,80 DM

○ auf Diskette zum Subscriptionspreis von 19,80 DM

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift
Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Unterschrift ____________________
(Ein Verrechnungsscheck/quittierter Einzahlungsbeleg liegt bei)

Screen-Display

Wenn Sie die Demonstration innerhalb von INPUT 64 gesehen haben, werden Sie sich sicherlich fragen, wie Sie ähnliche Efekte erreichen können.

Die Ansteuerung von BASIC aus ist denkbar einfach. Zuerst müssen Sie allerdings das Programm mit CTRL und S auf Ihren Datenträger sichern. Diese Version (sie enthält nicht die Demonstration) starten Sie einfach mit RUN.

Ihnen stehen jetzt zwei SYS-Aufrufe und eine "neue" POKE-Adresse zur Verfügung.

Mit dem ersten SYS-Aufruf legen Sie jeweils die Ausmaße des Fensters fest

Syntax = SYS 51200, von Zeile, bis Zeile, von Spalte, bis Spalte

und mit dem zweiten SYS-Aufruf können Sie jeweils eine Ebene "zurückblättern".

Syntax = SYS 51449 (ohne Parameter)

Nach jeder Fensterdefinition beziehen sich die folgenden PRINT-Anweisungen nur noch auf dieses Fenster. Sie können insgesamt 8 Fensterebenen definieren.

Nun kann es sinnvoll sein, eine bestimmte Fensterebene direkt anzuspringen. Hierfür gibt es eine POKE-Adrese.

Syntax = POKE 52224, Ebene

Die Ebene 0 ist hierbei der Original-Bildschirm, die Ebenen 1 bis 8 sind die eigentlichen Fenster (in der Reihenfolge, in der Sie angelegt wurden).

Wir wollen Ihnen auch nicht verschweigen, wo die Fenster aufbewahrt werden. Es gibt jeweils unter dem ROM und unter dem KERNAL einen 8KByte großen RAM-Bereich, der von BASIC aus nicht genutzt werden kann. Für einen Bildschirm werden 2KByte Speicher benötigt (jeweils 1KByte für die Zeichen und die Farbe - auch die wird nämlich "gerettet"), so daß also genau für 8 Bildschirme Platz vorhanden ist.

Damit der BASIC-Bereich nicht eingeschränkt werden muß, liegt das eigentliche Maschinen-Programm im Bereich von $C000 aufwärts.

Eine Anmerkung zum Schluß: Wenn Sie bei eingeschaltetem Fenster das Programm mit RUN/STOP (RESTORE) verlassen, sollten Sie mit einem POKE 52224,0 den Zeiger korrigieren.

SID-Kurs

Nachdem es in den ersten beiden Teilen der Serie um die wichtigsten klangtheoretischen Grundlagen der Musikprogrammierung ging, steht diesmal Praxis an. Die im Kurs erklärte Notenleseroutine kann auf eigenen Datenträger abgespeichert werden. Die gewünschten Noten müssen Sie dann ab Zeile 510 eingeben.

Übrigens: Nur diese Routine ist "save"bar, der Kurs insgesamt, wie alle INPUT 64-Serien, nicht!

In den bisherigen Kursteilen ging es um:

Ton, Frequenz, Lautstärke (Teil 1 ,INPUT 64 Ausgabe 6/85))
Wellenformen und Hüllkurven (Teil 2, INPUT 64 Ausgabe 7/85))

Im Beiheft der Ausgabe 7/85 befindet sich eine komplette Übersicht der bislang behandelten Sound-Chip-Adressen.

Listing Notenleseroutine

```
5 rem initialisieren
10 forx=0to24:poke54272+x,0:nextx
15 rem lautstaerke setzen
20 poke54272+24,15
25 rem huellkurve festlegen
27 poke54276,1*16+4:poke54277,6*16+10
30 rem notenleseroutine
100 read t1,t2,l
105 rem key-bit setzen und noten poken
110 poke54276,33:poke54272,t1:poke54273,t2
115 rem zeitschleife , key-bit loeschen
120 for w=1 to l:next w:poke54276,32
125 rem ende abfrage
130 if l<>251 then 100
500 rem datas
510 data 180,8,251
1000 rem die rem-zeilen fehlen im
1010 rem "savebaren" programm
1020 rem ansonsten ist es identisch

ready.
```

KLICKER

Eine Kugel, mittels Joystick oder Tastatur gesteuert, muß zu einem ZIEL gerollt werden. Ganz einfach, werden Sie sagen? Nun ja, aber kommen Sie erstmal über beziehungsweise durch die Hindernisse.

Tastaturbelegung:	Z	= Links
	X	= Rechts
	CRSR UP	= Hoch
	CRSR DOWN	= Runter

Das Spiel ist für maximal zwei Spieler ausgelegt. Es sind drei Runden zu überstehen. In der ersten Runde bestehen die Hindernisse aus Gittern, die die Kugel beschleunigen beziehungsweise abbremsen.

In der zweiten Runde geht es um symbolische Türen, die durch Berühren der Schlüssel aufgeschlossen werden müssen. Außerdem werden Sie von einem Hammer verfolgt, der Ihnen den Garaus machen will.

In der dritten Runde wird es ganz heikel. Sie werden zwar nicht verfolgt, dafür werfen aber die Gitter aus der ersten Runde neue Probleme auf. In dieser Runde wird Ihre Kugel nicht nur beschleunigt oder abgebremst, sondern wie mit einem Gummiband zurückgeschleudert. Achten Sie auch kurz vor dem Ziel auf die Wände, denn einige gekennzeichnete Stellen wirken wie ein Trampolin.

Natürlich hat der gewonnen, der die meisten Punkte hat.

Hilfsprogramme

Wir stellen Ihnen diesmal drei Routinen vor:

Mit der ersten können Sie einen 16bit-Wert in zwei aufeinander folgenden Adressen speichern, mit der zweiten einen entsprechenden Wert auslesen, und die dritte Routine nimmt Ihnen die mühselige Umrechnung der Zahlensysteme ab.

Sie kennen sicherlich das Problem, daß Sie Zeiger (zum Beispiel in der Zero-Page für den BASIC-Anfang die Adressen 43 und 44) verändern wollen. Hierfür sind einige Berechnungen notwendig. Nehmen wir an, Sie wollen

Sie müssen zuerst die Trennung in LOW- und HIGH-Byte vornehmen:

LOW -Byte = 3000-256*INT(3000/256)
HIGH-Byte = INT(3000/256)

Als Ergebnis bekommen Sie für das LOW-Byte 184 und für das HIGH-Byte 11.Diese Werte werden jetzt in die entsprechenden Adressen geschrieben. (POKE 43,184:POKE 44,11).

Soweit sogut und wahrscheinlich auch bekannt, nicht desto weniger jedoch recht aufwendig. Einfacher geht es nun mit unserem DOKE-Befehl.

SYS 51024,43,3000

Wobei 51024 die Einsprung-Adresse für die Routine ist, 43 die erste der beiden aufeinanderfolgenden Ziel-Adressen und 3000 der dezimale Wert, der diesen beiden Ziel-Adressen zugeordnet werden soll. (Da diese Routine immer zwei Adressen den Wert zuordnet, ist die größte Zahl, die hier stehen,darf 65535.)

Das Gegenstück zu dem DOKE-Befehl stellt die DEEK-Funktion dar. Mit Ihr können Sie einen 16bit-Wert aus zwei benachbarten Speicheradressen auslesen.

SYS 51058,43,A

ordnet der Variablen "A" den Wert aus der Adresse 43 und 44 zu.Beachten Sie aber bitte, daß DEEK genau wie PEEK kein Befehl sondern eine Funktion ist.

Abschließend wollen wir Ihnen noch den Befehl für die Zahlenbereichs-Umrechnung erklären.

Wenn Sie beispielweise in einem "ROM-Listing" HEX-Zahlen finden, bei der Programmerstellung für einen SYS-Aufruf aber den entsprechenden dezimalen Wert brauchen, brauchen Sie in Zukunft nur noch einzugeben:

SYS 51580, $ABCD

und erhalten als Ergebnis:

DEZ = 43981
HEX = $ABCD
BIN = %10101011 11001101

Sie erkennen bei dieser Ausgabe schon, wie die unterschiedlichen Zahlensysteme gekennzeichnet werden. (HEX = $, BIN = % und DEZ "ohne") Wenn Sie diese Kennzeichnung beachten, können Sie bei der Eingabe jedes der drei Zahlensysteme benutzen.

```
profi-ass 64 v2.0    seite 1

                   ;*********************************
                   ;***   d e e k   +   d o k e   ***
                   ;*********************************
                   ;
                   ; (c)  i n p u t   6 4   /   ausgabe 8/85
                   ; by frank boerncke
                   ;

c750                        *=    $c750   ; programm-start
c750                        .pag 66
c750                        .opt p4
aefd               chkcom   =     $aefd
ad8a               frmnum   =     $ad8a
b7f7               getadr   =     $b7f7
                   ;
                   ;sys51024,adresse,16bit-wert
                   ;
                   ;schreibt 16bit-werte in den speicher
                   ;
                   ;
c750 20 fd ae doke          jsr   chkcom  ;gibt syntax error, wenn kein komma
c753 20 8a ad               jsr   frmnum  ;16bit-wert von der adresse in fac
c756 20 f7 b7               jsr   getadr  ;fac nach $14/$15, akku und y-reg.
c759 84 71                  sty   $71     ;l-byte der adresse  nach $71
c75b 85 72                  sta   $72     ;h-byte der adresse nach $72
c75d 20 fd ae               jsr   chkcom  ;gibt syntax error  wenn kein komma
c760 20 8a ad               jsr   frmnum  ;16bit-wert von dem  oper. in's fac
c763 20 f7 b7               jsr   getadr  ;fac nach $14/$15, akku und y-reg.
c766 a2 00                  ldx   #$00    ;x-reg. zum doken auf null setzen
c768 98                     tya           ;l-byte vom wert in akku kopieren
c769 81 71                  sta   ($71),x  ;l-byte von wert in adresse poken
c76b e6 71                  inc   $71      ;adresszeiger um 1 erhoehen
c76d a5 15                  lda   $15      ;h-byte vom wert in akku laden
c76f 81 71                  sta   ($71),x  ;h-byte von wert in adresse poken
c771 60                     rts           ;zurueck zum basic

                   ;
                   ;sys51058,adresse,variable
                   ;
                   ;uebergibt 16bit-wert an variable
                   ;
                   ;
c772 20 fd ae deek          jsr   chkcom  ;wenn kein komma, dann syntax error
c775 20 8a ad               jsr   frmnum  ;16-bit wert von der adresse in fac
c778 20 f7 b7               jsr   getadr  ;fac nach $14/$15, akku und y-reg.
c77b a2 00                  ldx   #$00    ;x-reg. zum deeken auf null setzen
c77d a1 14                  lda   ($14),x  ;akku mit l-byte des wertes laden
c77f 85 63                  sta   $63      ;und dann wert in $63 merken
c781 e6 14                  inc   $14      ;adresszeiger um 1 erhoehen
c783 a1 14                  lda   ($14),x  ;akku mit h-byte des wertes laden
c785 85 62                  sta   $62      ;und dann wert in $62 merken
c787 a2 90                  ldx   #$90    ;vorbereitungen fuer die
c789 38                     sec           ;'jsr $bc49' routine
c78a 20 49 bc               jsr   $bc49   ;integerwert aus $62/$63 in's fac
c78d 20 fd ae               jsr   chkcom  ;gibt syntax error, wenn kein komma
c790 20 8b b0               jsr   $b08b   ;variableadresse in akku und x-reg.
c793 85 49                  sta   $49     ;merkt l-byte von adresse in $49
c795 84 4a                  sty   $4a     ;merkt h-byte von adresse in $4a
c797 20 90 ad               jsr   $ad90   ;gibt'type mismatch' bei stringv.
c79a 20 d0 bb               jsr   $bbd0   ;kopiert das fac in die variable
c79d 60                     rts           ;zurueck zum basic
üc750-c79e
no errors
```

```
profi-ass 64 v2.0    seite 1

                   ;*********************************
                   ;*** hex/dez/bin-umwandlung    ***
                   ;*********************************
                   ;
                   ; (c) i n p u t   6 4   /   ausgabe 8/85
                   ; by olaf krumrey + joerg viola
                   ;
c7a0                        *=    $c7a0   ; programm-start
c7a0                        .pag 66
c7a0                        .opt p4
0014               erg      =     $14          ; interne
00fd               mem      =     $fd          ; zwischenspeicher
                   ;
                   ; die folgenden routinen wandeln
                   ; nach internformat ($14)
                   ;
                   ; bin -> intern
                   ;
c7a0 a0 00         binint   ldy   #$00    ; buf2-pointer
c7a2 ca            loop2    dex           ; x auf ende buf
c7a3 bd b6 c9               lda   buf,x        ; a=bin-ziffer
c7a6 c9 31                  cmp   #"1"
c7a8 f0 0d                  beq   mark
c7aa c9 30                  cmp   #"0"         ; 0 oder 1 print
c7ac f0 09                  beq   mark         ; dann zu mark
c7ae c9 25                  cmp   #"%"         ; % print
c7b0 f0 26                  beq   rest         ; dann aufhoeren
c7b2 a2 0b         syn      ldx   #$0b
c7b4 4c 3a a4               jmp   $a43a        ; sonst syntax-err
c7b7 99 d6 c9      mark     sta   buf2,y  ; ziffer nach buf2
c7ba c8                     iny           ; inc pointer
c7bb 98                     tya
c7bc c9 10                  cmp   #$10         ; buf2 voll print
c7be 90 e2                  bcc   loop2        ; nein, weiter
c7c0 ca            loop3    dex
c7c1 bd b6 c9               lda   buf,x        ; naechste ziffer
c7c4 c9 25                  cmp   #"%"         ; % print
c7c6 f0 1b                  beq   los          ; dann umwandeln
c7c8 c9 30                  cmp   #"0"         ; 0 print
c7ca f0 f4                  beq   loop3        ; dann weiter
c7cc c9 31                  cmp   #"1"         ; 1 print
c7ce f0 03                  beq   ill          ; dann ill quan.
c7d0 4c b2 c7               jmp   syn          ; sonst syntax-err
c7d3 a2 0e         ill      ldx   #$0e     ; illegal quanti-
c7d5 4c 3a a4               jmp   $a43a        ; ty error
c7d8 a9 30         rest     lda   #"0"
c7da 99 d6 c9               sta   buf2,y
c7dd c8                     iny
c7de 98                     tya           ; rest von buf2
c7df c9 10                  cmp   #$10         ; mit 0 voll-
c7e1 d0 f5                  bne   rest         ; schreiben
c7e3 a2 00         los      ldx   #$00     ; die binzahl liegt
                   ;                 ; formatiert in buf2
c7e5 bd d6 c9      loop4    lda   buf2,x  ; a=ziffer
c7e8 38                     sec
c7e9 e9 30                  sbc   #"0"         ; asc"0"->0
c7eb 4a                     lsr
c7ec 66 14                  ror   erg          ; bit0 a->bit7 erg
c7ee e8                     inx           ; naechste ziffer
c7ef 8a                     txa
c7f0 c9 08                  cmp   #$08         ; 8. ziffer print
c7f2 d0 f1                  bne   loop4        ; nein, loop4  ja, erglow fertig
c7f4 bd d6 c9      loop5    lda   buf2,x
c7f7 38                     sec
```

```
profi-ass 64 v2.0    seite 2

1650:  c7f8 e9 30           sbc  #"0"
1660:  c7fa 4a              lsr
1670:  c7fb 66 15           ror  erg+1
1680:  c7fd e8              inx
1690:  c7fe 8a              txa
1700:  c7ff c9 10           cmp  #$10          ; dasselbe fuer
1710:  c801 d0 f1           bne  loop5         ; erghigh
1720:  c803 60              rts
1730:                ;
1740:                ; intern -> bin
1750:                ;
1760:  c804 a2 01    intbin ldx  #$01
1770:  c806 b5 14           lda  erg,x         ; a=erghigh
1780:  c808 c9 00           cmp  #$00          ; a=0 print
1790:  c80a f0 08           beq  low           ; dann nicht,
1800:  c80c 20 1a c8        jsr  out           ; sonst ausgeben
1810:  c80f a9 20           lda  #" "
1820:  c811 20 d2 ff        jsr  $ffd2         ; " " ausgeben
1830:  c814 a2 00    low    ldx  #$00
1840:  c816 20 1a c8        jsr  out           ; erglow ausgeben
1850:  c819 60              rts
1860:                ;
1870:  c81a b5 14    out    lda  erg,x     ; a=byte
1880:  c81c 85 fd           sta  mem           ; zwischenspeicher
1890:  c81e a2 00           ldx  #$00
1900:  c820 06 fd    loop6  asl  mem
1910:  c822 a9 00           lda  #$00
1920:  c824 2a              rol
1930:  c825 18              clc
1940:  c826 69 30           adc  #"0"          ; a="1" oder "0"
1950:  c828 20 d2 ff        jsr  $ffd2         ; ausgabe
1960:  c82b e8              inx
1970:  c82c 8a              txa
1980:  c82d c9 08           cmp  #$08          ; 8 bit ausgeben
1990:  c82f d0 ef           bne  loop6
2000:  c831 60              rts
2010:                ;
2020:                ; intern -> dez
2030:                ;
2040:  c832 a5 15    intdez lda  erg+1     ; a=erghigh
2050:  c834 a6 14           ldx  erg           ; x=erglow
2060:  c836 20 cd bd        jsr  $bdcd         ; rom-routine
2070:  c839 60              rts
2080:                ;
2090:                ; dez -> intern
2100:                ;
2110:  c83a 20 5b c8 dezint jsr  juhu
2120:  c83d a5 7a           lda  $7a
2130:  c83f 48              pha
2140:  c840 a5 7b           lda  $7b           ; programmzeiger
2150:  c842 48              pha           ; retten
2160:  c843 a9 b5           lda  #<(buf-1)
2170:  c845 85 7a           sta  $7a
2180:  c847 a9 c9           lda  #>(buf-1)     ; programmzeiger
2190:  c849 85 7b           sta  $7b           ; auf buf
2200:  c84b 20 73 00        jsr  $0073
2210:  c84e 20 8a ad        jsr  $ad8a
2220:  c851 20 f7 b7        jsr  $b7f7         ; buf ausgeben
2230:  c854 68              pla
2240:  c855 85 7b           sta  $7b
2250:  c857 68              pla           ; programmzeiger
2260:  c858 85 7a           sta  $7a           ; laden
2270:  c85a 60              rts
2280:                ;
```

```
profi-ass 64 v2.0    seite 4

2870:  c8d5 ad d6 c9         lda  buf2
2880:  c8d8 20 a6 c8         jsr  char1
2890:  c8db 0a               asl
2890:  c8dc 0a               asl
2890:  c8dd 0a               asl
2890:  c8de 0a               asl
2900:  c8df 65 15            adc  erg+1        ; ebenso high-byte
2910:  c8e1 85 15            sta  erg+1        ; erzeugen
2920:  c8e3 60               rts
2930:                 ;
2940:  c8e4 a9 30     hexint .lda  #"0"
2950:  c8e6 8d d6 c9         sta  buf2
2960:  c8e9 8d d7 c9         sta  buf2+1
2970:  c8ec 8d d8 c9         sta  buf2+2
2980:  c8ef 8d d9 c9         sta  buf2+3       ; buf2 loeschen
2990:  c8f2 ca               dex          ; ende buf
3000:  c8f3 a0 03            ldy  #$03         ; ende buf2
3010:  c8f5 bd b6 c9 loop9   lda  buf,x   ; a=ziffer
3020:  c8f8 c9 24            cmp  #"$"         ; ende print
3030:  c8fa f0 28            beq  end3         ; ja, dann wandeln
3040:  c8fc 86 fd            stx  mem
3040:  c8fe 48               pha          ; ziffer 0-f print
3050:  c8ff 20 a6 c8         jsr  char1        ; nein, syntax-err
3060:  c902 68               pla
3060:  c903 a6 fd            ldx  mem      ; nicht wandeln
3070:  c905 99 d6 c9         sta  buf2,y       ; zeichen->buf2
3080:  c908 ca               dex
3090:  c909 88               dey          ; naechste ziffer
3100:  c90a c0 ff            cpy  #$ff         ; ende print
3110:  c90c d0 e7            bne  loop9        ; nein, dann loop9
3120:  c90e bd b6 c9 loop10  lda  buf,x   ; a=ziffer
3130:  c911 c9 24            cmp  #"$"         ; ende print
3140:  c913 f0 0f            beq  end3         ; ja, dann wandeln
3150:  c915 ca               dex
3160:  c916 86 fd            stx  mem          ; ziffer speichern
3170:  c918 20 a6 c8         jsr  char1        ; und wandeln
3180:  c91b a6 fd            ldx  mem          ; wieder laden
3190:  c91d c9 00            cmp  #$00         ; ziffer=0 print
3200:  c91f f0 ed            beq  loop10       ; ja, dann loop10
3210:  c921 4c d3 c7         jmp  ill          ; nein, dann ill
3220:  c924 20 b7 c8 end3    jsr  char4   ; buf2 wandeln
3230:  c927 60               rts
3240:                 ;
3250:                 ; gesamtausgabe
3260:                 ;
3270:  c928 44 45 5a outtab  .asc "dez= hex=$bin=%"
3280:  c937 a9 0d     prout  lda  #$0d
3290:  c939 20 d2 ff         jsr  $ffd2        ; return ausgeben
3300:                 ;
3310:                 ; dezimalausgabe
3320:                 ;
3330:  c93c a2 00     dezout ldx  #$00
3340:  c93e bd 28 c9 loop13  lda  outtab,x
3350:  c941 20 d2 ff         jsr  $ffd2
3360:  c944 e8               inx
3370:  c945 e0 05            cpx  #$05         ; "dez=$"
3380:  c947 d0 f5            bne  loop13       ; ausgeben
3390:  c949 20 32 c8         jsr  intdez       ; zahl ausgeben
3400:  c94c a9 0d            lda  #$0d
3410:  c94e 20 d2 ff         jsr  $ffd2        ; return ausgeben
3420:                 ;
3430:                 ; hexadezimalausgabe
3440:                 ;
3450:  c951 a2 05     hexout ldx  #$05
```

```
profi-ass 64 v2.0    seite 3

2290:  c85b a2 00     juhu     ldx  #$00
2300:  c85d bd b6 c9  loop11   lda  buf,x
2310:  c860 c9 0d              cmp  #$0d
2320:  c862 f0 0f              beq  tend
2330:  c864 c9 3a              cmp  #$3a          ; "9"+1
2340:  c866 b0 08              bcs  tsyn
2350:  c868 c9 2f              cmp  #$2f          ; "0"-1
2360:  c86a 90 04              bcc  tsyn
2370:  c86c e8                 inx
2380:  c86d 4c 5d c8           jmp  loop11        ; testet jedes
2390:  c870 4c b2 c7  tsyn     jmp  syn      ; zeichen auf
2400:  c873 60       tend     rts           ; dezimalziffer
2410:                 ;
2420:                 ; intern -> hex
2430:                 ;
2440:  c874 30 31 32  tab      .asc "0123456789abcdef"
2450:  c884 aa        out1     tax           ; druckt a in hex
2460:  c885 bd 74 c8           lda  tab,x         ; a=ascii-ziffer
2470:  c888 20 d2 ff           jsr  $ffd2         ; ausgabe
2480:  c88b 60                 rts
2490:                 ;
2500:  c88c 48        out2     pha           ; gibt a aus
2510:  c88d 4a                 lsr
2510:  c88e 4a                 lsr
2510:  c88f 4a                 lsr
2510:  c890 4a                 lsr           ; obere 4 bit
2520:  c891 20 84 c8           jsr  out1          ; ausgeben
2530:  c894 68                 pla
2540:  c895 29 0f              and  #%00001111    ; untere 4 bit
2550:  c897 20 84 c8           jsr  out1          ; ausgeben
2560:  c89a 60                 rts
2570:                 ;
2580:  c89b a5 15     inthex   lda  erg+1    ; oberes byte
2590:  c89d 20 8c c8           jsr  out2          ; ausgeben
2600:  c8a0 a5 14              lda  erg           ; unteres byte
2610:  c8a2 20 8c c8           jsr  out2          ; ausgeben
2620:  c8a5 60                 rts
2630:                 ;
2640:                 ; hex -> intern
2650:                 ;
2660:  c8a6 a2 00     char1    ldx  #$00     ; asc->intern
2670:  c8a8 dd 74 c8  loop7    cmp  tab,x    ; a=tab print
2680:  c8ab f0 08              beq  found         ; ja, dann fertig
2690:  c8ad e8                 inx           ; naechste ziffer
2700:  c8ae e0 10              cpx  #$10          ; insgesamt 16
2710:  c8b0 d0 f6              bne  loop7         ; nein, loop7
2720:  c8b2 4c b2 c7           jmp  syn           ; sonst syntax-err
2730:  c8b5 8a        found    txa           ; a=intern
2740:  c8b6 60                 rts
2750:                 ;
2760:  c8b7 ad d9 c9  char4    lda  buf2+3
2770:  c8ba 20 a6 c8           jsr  char1         ; 4. zeichen
2780:  c8bd 85 14              sta  erg           ; wandeln
2790:  c8bf ad d8 c9           lda  buf2+2        ; 3. zeichen
2800:  c8c2 20 a6 c8           jsr  char1         ; wandeln
2810:  c8c5 0a                 asl
2810:  c8c6 0a                 asl
2810:  c8c7 0a                 asl
2810:  c8c8 0a                 asl           ; beide ergeben
2820:  c8c9 65 14              adc  erg           ; low-byte
2830:  c8cb 85 14              sta  erg
2840:  c8cd ad d7 c9           lda  buf2+1
2850:  c8d0 20 a6 c8           jsr  char1
2860:  c8d3 85 15              sta  erg+1
```

```
profi-ass 64 v2.0    seite 5

3460:  c953 bd 28 c9 loop14   lda  outtab,x
3470:  c956 20 d2 ff          jsr  $ffd2
3480:  c959 e8                inx
3490:  c95a e0 0a             cpx  #$0a          ; "hex=$"
3500:  c95c d0 f5             bne  loop14        ; ausgeben
3510:  c95e 20 9b c8          jsr  inthex        ; zahl ausgeben
3520:  c961 a9 0d             lda  #$0d
3530:  c963 20 d2 ff          jsr  $ffd2         ; return ausgeben
3540:                ;
3550:                ; binaerausgabe
3560:                ;
3570:  c966 a2 0a    binout   ldx  #$0a
3580:  c968 bd 28 c9 loop15   lda  outtab,x
3590:  c96b 20 d2 ff          jsr  $ffd2
3600:  c96e e8                inx
3610:  c96f e0 0f             cpx  #$0f          ; "bin=%"
3620:  c971 d0 f5             bne  loop15        ; ausgeben
3630:  c973 20 04 c8          jsr  intbin        ; zahl ausgeben
3640:  c976 a9 0d             lda  #$0d
3650:  c978 20 d2 ff          jsr  $ffd2         ; return ausgeben
3660:  c97b 60                rts           ; zum basic


3670:                ;
3680:                ;=== h a u p t p r o g r a m m ===
3690:                ;   sys 51580 einsprung-adresse
3700:                ;


3710:  c97c a2 00             ldx  #$00
3720:  c97e 20 79 00          jsr  $0079         ; chrgot, letztes
3730:  c981 c9 2c             cmp  #","
3740:  c983 f0 03             beq  loop16
3750:  c985 4c b2 c7          jmp  syn
3760:  c988 20 73 00 loop16   jsr  $0073    ; chrget
3770:  c98b 9d b6 c9          sta  buf,x         ; zeichen -> buf
3780:  c98e e8                inx           ; naechstes
3790:  c98f c9 00             cmp  #$00          ; zeilenende print
3800:  c991 d0 f5             bne  loop16        ; nein,dann loop16
3810:  c993 ca                dex           ; x auf return
3820:  c994 a9 0d             lda  #$0d          ; return in
3830:  c996 9d b6 c9          sta  buf,x         ; buf speichern
3840:  c999 ad b6 c9          lda  buf           ; erstes zeichen
3850:  c99c c9 24             cmp  #"$"          ; "$" print
3860:  c99e d0 06             bne  not1
3870:  c9a0 20 e4 c8          jsr  hexint        ; ja, dann hex
3880:  c9a3 4c 37 c9          jmp  prout         ; zur ausgabe
3890:  c9a6 c9 25    not1     cmp  #"%"     ; "%" print
3900:  c9a8 d0 06             bne  not2
3910:  c9aa 20 a0 c7          jsr  binint        ; ja, dann bin
3920:  c9ad 4c 37 c9          jmp  prout         ; zur ausgabe
3930:  c9b0 20 3a c8 not2     jsr  dezint   ; sonst dez
3940:  c9b3 4c 37 c9          jmp  prout         ; und zur ausgabe
3950:  c9b6          buf      *=   *+32     ;unformatierter
3960:  c9d6          buf2     *=   *+32     ;formatierter buffer
3970:                ;
3980:                ;  laenge
3980:  c9f6
3990:                ;  $0387 = 903 bytes
4000:                ;

üc7a0-c9f6
no errors
```

TEST MUSIK-KEYBOARDS

Um den zu testenden Musiksoftwarepaketen gerecht zu werden, muß einem Test zunächst eine Beschreibung und Beurteilung der Soundqualitäten des C64 vorangehen.

Das Herz der Tonerzeugung im 64er ist das Sound Interface Device, kurz SID genannt. In diesen wenigen Kubikmillimetern Silizium versteckt sich in vereinfachter Form das, womit Robert Moog seinen legendären Minimoog ausrüstete. Das Grundprinzip des Moog-Synthesizers bestand aus unabhängigen Moduln, die durch definierte Steuerspannungen kontrolliert werden und ihrerseits Steuerspannungen abgeben. Die Bezeichnungen der Moduln ,z. B. VCO für Voltage-controlled-Oscillator für einen Tonerzeuger, spiegelt diesen Sachverhalt wieder.

Dies bedeutet in der Praxis, das in Abhängigkeit von der Steuerspannung die Frequenz des Oszillators festgelegt wird. Je höher die Spannung, desto höher der Ton.

Baugruppen - Moduln - wie VCA für Lautstärke und VCF für Filter arbeiten entsprechend. Einmal wird mit der Steuerspannung die Lautstärke festgelegt (Voltage-controlled-Amplifier), das andere Mal die Filterstärke durch die Steuerspannung definiert.

Der SID besitzt zunächst einmal 3 Oszillatoren. Diese erzeugen elektrische Schwingungen in Form von Dreiecks-, Sägezahn- und Rechteckwellen. Bei der Rechteckwelle kann das Tastverhältnis, also das zeitliche Verhältnis von hohem zu tiefem Spannungspegel, verändert werden. Die drei Wellenformen besitzen ein unterschiedliches Obertonspektrum, sie haben jeweils spezielle Klangbilder. Außerdem können die Oszillatoren ein Rauschen erzeugen, das zur Simulation von Geräuschen wie z. B. Explosionen dient.

Der dritte Oszillator kann auch als LFO (Low-Frequency-Oscillator) benutzt werden. Steuert man mit seiner Frequenz die Lautstärke, so entsteht ein Vibrato-Effekt.

Ein zusätzlicher Filter formt den Klang dadurch, daß nur solche Anteile des Obertonspektrums den Filter passieren können, die im Klangbild erwünscht sind. Werden hohe Teiltöne weggefiltert, spricht man von einem Tiefpassfilter, bei Wegfiltern von tiefen Teiltönen von einem Hochpassfilter. Beide Arten können zu einem Bandpassfilter kombiniert werden, das ein Frequenzband im mittleren Bereich durchläßt. Mit dem Filter kann man also bestimmen, ob das Klangbild dumpf oder hell erscheinen soll.

Der Hüllkurvengenerator beeinflußt den zeitlichen Ablauf der Lautstärke oder des Klangs. Er läßt den Ton langsam oder schnell

anschwellen (Attack), abschwellen (Decay) und bestimmt, mit welchem Pegel der Ton ausgehalten wird (Sustain) und wie schnell er nach dem Loslassen der Taste verklingt (Release). Diese Funktionen sind sehr wichtig, weil wir viele Instrumente nicht nur nach dem Klangbild, sondern vorwiegend auch nach dem zeitlichen Verlauf des Klanges unterscheiden. Als Vergleich seien hier Trompete und Xylophon genannt.

Weitere Einzelheiten würden den Rahmen dieses Tests sprengen. Wenn Sie sich näher mit den Möglichkeiten der Musikprogrammierung mit Hilfe des SID vertraut machen wollen, empfehle ich Ihnen den SID-Kurs in diesem Magazin.

Der C64 ist also vom Prinzip her wie ein richtiger Synthesizer mit VCO, VCF, VCA, LFO und Hüllkurvengenerator ausgestattet.

Wenn Sie die musikalischen Qualitäten Ihres 64ers schon einmal direkt mit einem handelsüblichen Synthesizer verglichen haben, werden Sie jedoch ein wenig enttäuscht gewesen sein. Warum der C64 meist dünn und piepsig klingt, liegt einerseits an der Qualität der verwendeten Oszillatoren. Wenn schon das Grundmaterial, aus dem ein Klang gebastelt werden soll, nicht rund und voll klingt, kann man natürlich durch weitere Bearbeitung, die ja in diesem Fall aus Wegnehmen des nicht benötigten Klangmaterials besteht, nicht mehr an Klangfülle gewinnen.

Andererseits sind die Möglichkeiten der Klangbearbeitung ziemlich begrenzt. Selbst mit teuren Synthesizern ist eine naturgetreue Nachahmung von Musikinstrumenten recht schwierig, weil nur eine begrenzte Anzahl von VCO's, VCF's, VCA's und Hüllkurven zur Verfügung steht. Ein Synthesizer klingt immer noch nach Synthesizer und nicht nach Klavier, weil die subtilen Klangstrukturen nicht mit so ungenauem Werkzeug wie z. B. einem einzigen Tiefpaßfilter nachgearbeitet werden können.

Das gilt besonders für Synthesizer mit passiver Klangbearbeitung. Das sind solche, deren Klangerzeuger einige wenige Wellenformen und damit Grundklänge erzeugen, aus denen die benötigten Bestandteile herausgefiltert werden.

Seit einiger Zeit werden aber auch andere Tonerzeugungsprinzipien angewendet, die zu besseren Ergebnissen führen. Als Beispiel sei die additive Synthese genannt, die nach Fourier analysierte Wellenformen nachbildet. Das bedeutet, das ein Klang durch Zusammenfügen von Sinusschwingungen aufgebaut wird. Es gibt in der Mathematik Beweise für die Annahme, daß jeder noch so komplizierte Schwingungsverlauf bis zu einer gewissen Genauigkeit durch Überlagerungen von Sinusschwingungen nachgebildet werden kann.



W5444753 F
3000 Mark war
1000
DEUTSCHE BUNDESBANK
TAUSEND DEUTSCHE MARK
Hier n
Sie kö
W56161
und nat
Wichtig: Werfen Sie einen B
damit Ihr Programm auch in

Der

...ten auf den Gewinner!

Der Wettbewerb geht weiter.

...och einmal kurz die Bedingungen:
...nnen einsenden:

- Grafikprogramme
- Musikprogramme
- Spiele-
- Lernprogramme
- Anwenderprogramme

...ürlich völlig neue Programmideen.

...lick in das Kapitel "Hinweise für Autoren",
...nerhalb von INPUT 64 lauffähig ist.

... Rechtsweg ist wie immer ausgeschlossen.

Subtraktive (passive) Klangerzeugung bedeutet also:
- je mehr Veränderungen desto weniger Dynamik,

Additive Klangerzeugung dagegen:
- je mehr Veränderungen desto reichhaltiger der Ton.

Der populärste Vertreter dieser Synthesizerfamilie ist wohl der Yamaha DX7, der durch seine realitätsnahen Naturklänge verblüfft.

Aber selbst wenn man von der Nachahmung natürlicher Instrumente absieht, klingt der C64 im Vergleich zu richtigen Synthesizern etwas dünn. Es mangelt einfach am satten Grundklang der Oszillatoren.

Weiterhin fehlt ein zweiter Hüllkurvengenerator für den Filter, damit auch der zeitliche Ablauf der Obertonstruktur, der sich ja durchaus extrem von dem Lautstärkeverlauf unterscheiden kann, zu realisieren ist.

Zudem sind 3 Oszillatoren für ein ernstes polyphones Spiel ein bißchen wenig.

Ich will hier nicht den C64 wegen seiner Soundqualitäten verdammen. Schließlich handelt es sich hierbei nicht um ein Musikinstrument, sondern zunächst einmal um einen Computer, der innerhalb seiner Klasse von den Soundmöglichkeiten her Beachtliches leistet.

Vergleicht man ihn aber mit einem Musikinstrument, wird man Abstriche machen müssen. Diesen Hintergrund muß auch ein Testbericht über Musiksoftware für den C64 berücksichtigen.

Die Produkte, um die es in diesem Test geht, versuchen nicht wie einige andere Programme, dem Benutzer das Spielen eines Instrumentes, in diesem Falle eines Keyboards, abzunehmen. Hier werden keine Melodien in einen Sequenzer eingegeben, der dann vollautomatisch sein Musikprogramm abspult. Vielmehr kann der Benutzer Melodien oder Musikstückchen, die er im Kopf hat oder auf einem Notenblatt liest, mit den Fingern auf der Tastatur spielen. Die Musik wird also noch richtig mit der Hand gemacht. Das kann man zwar mit anderen Programmen auch, aber das Besondere an den Testobjekten ist eine Tastatur wie bei einem Keyboard, also weiße und schwarze Tasten in der bekannten Anordnung.

Die umständliche Orientierung auf der Computertastatur soll durch das problemlose Spiel auf einer gewohnten Klaviatur ersetzt werden. Damit ist auch die Zielsetzung klar: Der hochvergeistigte Computerfreak kann sich nach harter Programmierarbeit beim Spiel einer verträumten Melodie entspannen, ohne sich von seinem liebsten Spielzeug trennen zu müssen.

Das Umsetzen eines Notenbildes auf eine Klaviatur kann er wie bei einem normalen Instrument erlernen oder ausüben. Doch nun zum eigentlichen Test.

Besprochen werden hier das Wersiboard Music 64 und das incredible Keyboard aus dem Hause Sight & Sound.

Das Jen Musipack lag trotz wiederholter Anfragen beim Hersteller nicht rechtzeitig zum Testbeginn vor.

DAS WERSIBOARD MUSIC 64

Zum Test lag mir die Grundausführung der Software sowie die zugehörige Tastatur, das Wersiboard, vor. Mitgeliefert wurde eine einfache, übersichtliche Bedienungsanleitung mit Beispielen für die Soundgestaltung und einer Erläuterung der wichtigsten Synthesizerfunktionen.

Das Keyboard

Die Tastatur ist in einem Kunststoffgehäuse untergebracht, das in Form und Farbe an das Gehäuse des C64 angeglichen ist. Das Wersiboard ist leicht, handlich und kann sicherlich den ein oder anderen Stoß vertragen. Die Klaviatur hat einen Umfang von 4 Oktaven, die Tasten haben die normale Größe von Klaviertasten und bestehen aus Kunststoff. Dem erfahrenen Klavierspieler wird der Tastenanschlag nicht zusagen, wer aber häufig mit preiswerten Synthies und Orgeln umgeht, wird etwas labberige Tastaturen gewohnt sein.

Die Bespielbarkeit ist für diese Verhältnisse normal. Das Keyboard wird durch ein Flachbandkabel und einen Modulstecker mit dem Computer verbunden, der Anschluß ist unproblematisch.

Die Software

Die zugehörige Software ist wahlweise auf Kassette oder Diskette erhältlich. Die Diskettenversion, die mir zum Test vorlag, ließ sich problemlos laden. Zunächst erscheint auf dem Bildschirm ein Begrüßungstext, der auf mich etwas hausbacken wirkt. Wo andere Software-Hersteller mit Farben und ansprechender Graphik arbeiten, nutzt Wersi diesen Raum für eine Selbstdarstellung als weltbekannter bedeutender Orgelhersteller.

Nach diesem deutschsprachigen Text wird die weitere Benutzerführung erstaunlicherweise in englischer Sprache fortgesetzt.

Wersi bietet zwei Betriebsarten an:

Mono 64 und Poly 64.

Das Programm Poly 64 ermöglicht ein dreistimmiges Spiel, während bei Mono 64 jeweils nur ein Ton zur Zeit erklingt. Dem Spieler stehen verschiedene Klänge im Soundspeicher zur Verfügung, die er selbst modifizieren, neu entwickeln und auch auf Diskette abspeichern kann. Interne Speicherkapazität ist reichlich vorhanden, die Möglichkeit der Speicherung auf externem Datenträger setzt dem Sammeln von selbstentwickelten Klängen keine Grenzen.

Im Programm Poly 64 kann der Sound der 3 Stimmen nicht einzeln eingestellt werden, alle drei Stimmen klingen gleich. Über die Funktionstasten werden die Hüllkurvenparameter Attack, Decay, Sustain, Release sowie die Tonhöhe, Wellenform, Lautstärke und das Tastverhältnis für die Rechteckschwingung eingestellt. Die Graphik ist übersichtlich, die Bedienung unkompliziert. Die Klänge, die mit den wenigen zur Verfügung stehenden Parametern entwickelt werden können, sind allerdings nicht gerade vielseitig.

Mehr Soundmöglichkeiten bietet das Programm Mono 64. Allerdings ertönt hier immer nur der Ton der höchsten gedrückten Taste. Alle tiefer gespielten Tasten werden ignoriert. Auf dem Bildschirm erscheint ein Bedienfeld, in dem die eingestellten Parameter angezeigt werden.

Will ich einen Paramter verändern, bewege ich den Cursor mit Hilfe der Funktionstasten zu dem entsprechenden Parameter und erhöhe oder vermindere den eingestellten Wert.

Die Wellenform, Pulsweite der Rechteckschwingung und die Tonhöhe der zwei tonerzeugenden Oszillatoren können unabhängig voneinander gewählt werden.

Der zweite Oszillator kann gegenüber dem ersten leicht verstimmt werden, was wichtig für einen etwas fülligeren Klang ist. Diese beiden Oszillatoren können auch synchronisiert und ringmoduliert werden. Letzteres bewirkt einen glockenähnlichen Klang.

Der dritte Oszillator dient als LFO. Er bietet nur drei Wellenformen, nämlich Dreieck, Sägezahn und Rauschen. Er kann in der Geschwindigkeit reguliert werden. Weiterhin kann die Intensität, mit der der LFO auf den Filter und/oder die Tonhöhe einwirkt, gewählt werden.

Der Hüllkurvengenerator erzeugt im bekannten Sinne mit den vier Phasen Attack, Decay, Sustain und Release.

Der Filter bietet die drei Betriebsarten Hochpaß, Tiefpaß und Bandpaß, kann aber auch ganz abgeschaltet werden. Grenzfrequenz und

Resonanzüberhöhung bei der Grenzfrequenz sowie die Intensität, mit der der Hüllkurvengenerator auf den Filter wirkt, können ausreichend fein abgestuft werden.

Gestört hat mich in beiden Teilprogrammen ein unnötiger Tastendruck, der zwischen Spielmodus und Soundprogrammiermodus hin und her schaltet. Habe ich einen Soundparameter geändert, muß ich mit einem zusätzlichen Tastendruck in den Spielmodus umschalten, um die Veränderung zu hören.

Will ich nun den Klang weiter verändern, muß ich erst wieder eine Taste zum Umschalten in den Programmiermodus umschalten. Ein direktes Hören der Veränderung und direkter Zugriff auf die Klangparameter ist so nicht möglich.

Insgesamt läßt die Bedienungssicherheit sehr zu wünschen übrig. So ist die Bildschirmgraphik mit Hilfe der Tastaturgraphikzeichen aufgebaut, aber die Umschaltung von Großschrift auf Kleinschrift ist nicht verriegelt. Drückt man die Commodoretaste und die Shifttaste gleichzeitig und schaltet so auf Kleinschrift um, ist die übersichtliche Graphik hin. Das ist zwar durch erneute Umschaltung reparabel, aber doch lästig. Das gleich gilt für die Rückkehr ins Basic über die Betätigung der Run/Stop-Taste. Will ich ein Soundpreset aus einer Speicherbank laden oder in diese schreiben und gebe statt einer Zahl einen Buchstaben ein, werden Buchstaben und Graphikzeichen davor gelöscht, sodaß die Rahmen um das betreffende Anzeigefeld nun wirklich im Eimer sind. Diese Panne kann nur über Rückkehr in das Basic und erneutes Starten des Programms korrigiert werden.

Es kommt vor, daß bei schnellem Spiel im Poly-Programm der dritte Ton beim Tastenwechsel verschluckt wird. Das gleiche passiert im Mono-Programm, hat aber hier fatalere Folgen, weil der Ton gänzlich ausbleibt, solange man die betreffende Taste gedrückt hält. Hebt man den Finger und spielt den Ton erneut, erklingt der Ton wieder. Das darf eigentlich bei einem normalen Keyboardinstrument nicht passieren.

Für die Qualität der Sounds, die man mit Hilfe der ganzen Anordnung erzeugen kann, gilt das eingangs Gesagte. Für einen Computer ist die Klangfülle beachtlich, für ein Musikinstrument dürftig. Was im C64 an Klangmöglichkeiten drinsteckt, ist zwar bei weitem noch nicht ausgeschöpft, aber die Klangmöglichkeiten sind auch dann gemessen an einem echten Synthesizer unbefriedigend. Beim Zusammenklang der drei Stimmen im Poly-Modus kratzt der Ton oft unangenehm, wenn nicht gerade Quinten oder Quarten gespielt werden. Wird kein Ton gespielt, nervt das verbleibende Brummen und Sirren aus dem Lautsprecher.

Durch das Keyboard wird die Illusion geweckt, es handele sich bei der ganzen Apparatur um ein Musikinstrument. Diese Erwartung wird aber durch die klanglichen Möglichkeiten enttäuscht.

SIGHT & SOUND INCREDIBLE KEYBOARD

Der Lieferumfang des incredible Keyboard umfaßt eine Aufsatztastatur für die Tastatur des C64, eine Programmkassette, eine sehr sparsame, aber ausreichende Bedienungsanleitung und zwei Songbücher mit alten und modernen Songs in Melodie, Text und Begleitharmonien.

Das Keyboard

Bei dem Aufsatzkeyboard handelt es sich um einen Rahmen, der ziemlich genau auf die Computertastatur paßt und in Farbe und Form an das Commodore-Design angelehnt ist. Die wichtigen Bedienungselemente wie die Funktionstasten sind durch Aussparungen im Rahmen zugänglich.

An diesem Rahmen ist eine verkleinerte Klaviertastatur mit einem Tonumfang von zwei Oktaven beweglich befestigt. Über eine Art Stößel werden die unter der Klaviatur liegenden Tasten der Computertastatur beim Spielen auf dem Keyboardaufsatz niedergedrückt. Im Prinzip kann man also genausogut auch auf der Computertastatur direkt spielen, aber über das kleine Aufsatzkeyboard kann der Spieler leichter die richtigen Töne finden, wenn er mit einer Klaviertastatur vertraut ist, oder die Verbindung zwischen gedrückter Klaviertaste und klingendem Ton erlernen. Das ganze macht einen ganz putzigen Eindruck auf mich. Die verkleinerte Tastatur ist richtig niedlich, wie ein Puppenklavier.

In der beiliegenden Bedienungsanweisung wird wiederholt eindringlich davor gewarnt, die ganze Konstruktion an den Tasten hochzuheben, weil diese flexibel sein müssen und deshalb leicht abbrechen können.

Beim Spielen auf der Klaviatur hatte ich allerdings Schwierigkeiten, weniger wegen der kleinen Tasten, sondern weil der ganze Aufsatz wie ein Lämmerschwanz wackelte. Der obere Rand des Rahmens soll laut Bedienungsanleitung in eine der Rillen im Computergehäuse eingerastet werden, die sich oberhalb der Tastatur befinden. Da aber der Rahmen des Aufsatzes nicht ganz gerade war, sprang er immer wieder aus der Nut heraus.

Mögliche Abhilfe: Eine Hand hält den Rahmen, die andere spielt, oder einfach ein Streifen Klebeband.

Beide Maßnahmen beseitigen aber nur eine konstruktionsbedingte Schwäche des Aufsatzes. Wäre der Rahmen nicht leicht verzogen, wäre wohl ein festerer Sitz möglich.

Software

Im Vergleich zur Wersiboard-Software handelt es sich beim incredible Keyboard um eine Spar-Version.

Die Möglichkeiten der Klangbeeinflussung reduzieren sich:
- auf das Einstellen der Lautstärke,
- die Auswahl der Tonhöhe in Oktaven,
- der Wahl zwischen den drei Instrumenten Synthi, Piano und Baß.

Spielt man den Baß in höherer Tonlage, klingt der Ton eher wie ein Cembalo.

Ein Vibrato und ein Glide-Effekt können alternativ zugeschaltet werden. Über die Shift-Tasten kann ein Ausklingen des Tones auch nach Loslassen der Taste ähnlich wie bei einem Klavierpedal erzeugt werden.

Mit der Space-Taste kann man den Ton nach oben ziehen, das heißt, die Tonhöhe gleitet nach oben, solange die Space-Taste gedrückt bleibt.

Der Spielraum für Klangexperimente ist also äußerst eingeschränkt. Das Keyboard ist dreistimmig spielbar, solange kein Vibrato zugeschaltet ist.

Kritik muß ich auch hier am gelegentlichen Verschlucken von Tönen üben. Außerdem kommt es bei eingeschaltetem Glide-Effekt gelegentlich zu unbeabsichtigten Wiederholungen. Der Ton wirkt wie durch einen LFO mit Sägezahnkurve moduliert.

Die klanglichen Qualitäten der drei Instrumente sind nicht berauschend. Hier werden aber, anders als beim Wersiboard, durch den gesamten Spielzeugcharakter auch keine falschen Erwartungen geweckt.

Eine nette Geste für den Nichtmusiker sind die sehr einfach aufgebauten Songbooks. In einem befindet sich am Ende ein Abziehfolie mit der Bezeichnung der Töne. Diese Aufkleber kann der Nichtnotist als Orientierung auf die Tasten der Klaviatur kleben, um so zunächst über die Buchstabenbezeichnungen der Töne die Klaviatur kennenzulernen.

Fazit

Wer ernsthaft musizieren will, wird schnell von beiden Produkten ablassen, weil die klanglichen Möglichkeiten des Computers nicht mit einem Musikinstrument mithalten können. Wer einfach ein bißchen herumklimpern oder Noten lernen möchte, ist mit dem incredible Keyboard ganz gut bedient. Zum Kennenlernen und Entwickeln der Klangmöglichkeiten seines Computers kann sich der interessierte Nichtmusiker mit dem Wersiboard nett die Zeit vertreiben.

Sicherlich sind beide Systeme entwicklungsfähig, wie das Demo für den Sight & Sound Rythm Rocker auf der Programmkassette oder die teurere Aufbauversion der Wersiboard-Software erahnen lassen.

Das incredible keyboard liegt im Preis um 150,- DM, für das Wersiboard mit der Grundsoftware muß man schon über 400,- DM berappen.

Für Nachzügler und "Spätzünder"

Wegen der großen Nachfrage haben wir bereits vergriffene Ausgaben von INPUT 64 nachproduziert, so daß alle bisher erschienenen Ausgaben wieder lieferbar sind!! Ab Ausgabe 4/85 ist INPUT 64 auch auf Diskette erhältlich. Richten Sie Ihre Bestellung bitte direkt an den Verlag; die Lieferung erfolgt nur gegen Vorkasse (Verrechnungsscheck).
Kassettenversion: DM 12,80;
Diskettenversion: DM 19,80.
(incl. Porto und Verpackung)



INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

ÜBRIGENS: Abonnenten sparen den Preis von zwei INPUT 64 - Ausgaben pro Jahr (Diskettenversion). Und zwar ohne den üblichen "Pferdefuß", denn das INPUT 64 - Abo ist jederzeit mit Wirkung ab der übernächsten Ausgabe kündbar!

Aus dem Inhalt: Ausgabe 1 - Dateiverwaltung, drei (!) Spiele * Ausgabe 2 - Textprogramm, Zeichengenerator * Ausgabe 3 - Spriteeditor, Maschinensprache-Monitor * Ausgabe 4 - SuperTape DII, Grafikhilfe, Urlaubskalender * Ausgabe 5 - Mathe mit Nico, Talk to me (Dialogsimulation), Hintergrundmonitor * Ausgabe 6 - Haushaltsbuchführung, Recorder-Justage, Textadventure, BASIC-Compactor * Ausgabe 7 - Scroll Editor, TapeCopy, HiFi-Boxenberechnung.

HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise:

Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:

Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter.

Das Programm darf nur in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden.

Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein.

Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht.

Senden Sie uns Ihre Programme bitte auf Kassette oder Diskette mit kommentiertem Listing und Beschreibung.

Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlich erschwert wurde, zukünftig ungeprüft zurücksenden.

Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse:

1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0800-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF).

2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein.

3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein.

4. Die CTRL-Taste darf nicht benutzt werden.

Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

DiscMon

Nachdem in Ausgabe 6/85 ein speziell auf Kassettenbenutzer zugeschnittenes Programm den 3000 DM-Wettbewerb gewonnen hat, kommen diesmal die Diskettenbesitzer auf ihre Kosten. Der Diskmonitor von Tim Pritlove, der komfortabel die direkte Modifikation der Disketteninformation erlaubt, ist eines der intelligentesten Tools dieser Art auf dem Markt. Und daß einem diese Intelligenz nächtelange Arbeit und den Verlust wertvoller Daten ersparen kann, weiß jeder Floppy-Benutzer, der sich schon einmal mit "unerklärlichen" Diskettenfehlern herumärgern mußte.

Der Autor über sich selbst:

"Mein Name ist Tim Pritlove. (Er spricht sich übrigens genauso englisch aus, wie er aussieht.) Geboren wurde ich am 25.11.1967.

Meine Computerlaufbahn klingt recht langweilig, wahrscheinlich ist sie es auch. Sie begann dort, wo sie schon für viele begann: im Kaufhaus, oder genauer: in sämtlichen Computerabteilungen meines Wohnortes Hannover. Nachdem ich mich gerade noch vor dem Erwerb eines ZX-81 retten konnte, stieg ich auf dem C64 ein und schrieb meine ersten sinnlosen Programme. Natürlich kam auch ich nicht um den Struktur- und Geschwindigkeitsschocker BASIC herum und geriet dann durch reines Interesse am Anderen zur Maschinensprache-Programmierung.

Der Discmonitor ist mein erstes Maschinenprogramm überhaupt, und wie es aussieht auch mein letztes. Zwischen den manchmal endlos erscheinenden Programmierpausen entstanden andere, aber nicht ähnliche Utilities. Den Discmonitor schrieb ich aus Ärger auf alle vergleichbaren Disketten-Hilfsprogramme, die mir allesamt nicht die Leistung offenbaren konnten, die ich von ihnen erwartete. Motto : Selber schreiben ist Trumpf!"

VON SPUREN, BLÖCKEN UND SEKTOREN

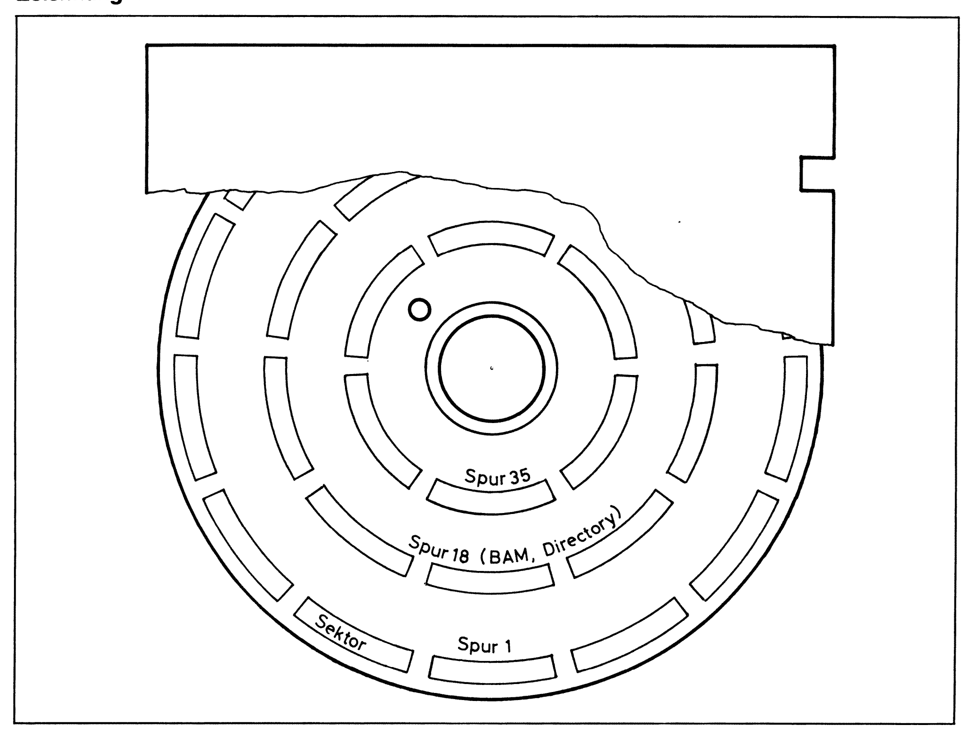
Die 5 1/4 Zoll-Disketten der 1541 werden beim Formatieren in Spuren (Tracks), und diese wiederum in Sektoren (Sectors) unterteilt. Die äußere Spur ist die Spur 1, die innere Spur 35. (Zeichnung 1) Die Sektoren jeder Spur sind von Null an durchnumeriert. Da der Umfang der Spuren von aussen nach innen abnimmt, ist die Anzahl der Sektoren pro Spur unterschiedlich. (Tabelle 1).

Tabelle 1

Spur		Sektoren	
dez.	hex	dez	hex
01 - 17	$01 - $11	0 - 20	$00 - $14
18 - 24	$12 - $18	0 - 18	$00 - $12
25 - 30	$19 - $1E	0 - 17	$00 - $11
31 - 35	$1F - $23	0 - 16	$00 - $10

Jeder dieser Sektoren stellt einen 'Block' dar, deren frei verfügbare Menge immer am Ende eines jeden Inhaltverzeichnisses angegeben wird (XXX BLOCKS FREE.). Das Inhaltsverzeichnis der 1541-Disketten befindet sich auf Spur 18. Zum Inhaltsverzeichnis gehört zum einen das ladt- und "list"bare Directory, in dem Programmnamen, Programmtyp, Programmlänge und der erste Track und Sektor der Programmdaten auf der Diskette angegeben ist. Der normalen Benutzeraugen meist nicht sichtbare Teil des Disketten-Inhaltsverzeichnisses ist die BAM. BAM steht für Block Available Map, zu deutsch Blockbelegungsplan. Die BAM befindet sich immer in Track 18/Sektor 0, zusammen mit einigen Informationen des Directorys, wie ID-Kennzeichnung der Diskette, Diskettenname und so weiter. In Sektor 1 dieser Spur beginnt dann das eigentliche Directory.

Zeichnung 1



Jeder Sektor kann 256 Bytes Daten aufnehmen. Zwei Bytes davon werden für die Blockverkettung, das heißt für den Zeiger auf den nächsten logischen Block, benötigt. Das erste Byte zeigt auf die Spur, das zweite auf den Sektor des nächsten zum jeweiligen Programm bzw. der Direktory gehörenden Datenblocks. Das Kennzeichen für den letzten Block ist eine Null im ersten Byte, das zweite Byte zeigt in diesem Fall die restliche Programmlänge in diesem Block an. (Eine Ausnahme bildet die Directory, hier steht das zweite Byte im letzten Sektor pauschal auf 255.)

Mit dem Discmonitor kann nun auf einfache Art jeder dieser Blöcke einzeln von Diskette in die 10 frei wählbaren Arbeitsspeicher (die 'm-Blocks' = Memory Blocks) gelesen und zurückgeschrieben werden.

BEFEHLSÜBUNGEN

Jeder Befehl wird durch einen Buchstaben dargestellt. Als Trennzeichen dient das Leerzeichen, es kann aber ausnahmslos auch weggelassen werden. Um einen Block zu lesen, benötigen wir das 'R', wie "READ". Da der Discmonitor über mehrere (10) m-Blocks verfügt, muß jeder direkt nach dem Befehl adressiert werden. Jeder m-Block wird durch eine Ziffer von 0 bis 9 dargestellt. Wollen wir also Track 18 Sektor 2 von Diskette in m-Block 4 lesen, so sieht das so aus: R4 12 02

Haben sie es bemerkt? Alle Parameter werden in hexadezimaler Form angegeben, aber das sollte man von einem Monitor gewohnt sein, denn der Vorteil ist nicht von der Hand zu weisen: alle Zahlen können durch zwei Zeichen dargestellt werden. Wer damit Schwierigkeiten hat, sollte schnellstens zu einer Konversionstabelle greifen oder sich des #-Befehls bedienen, mit dem man leicht dezimale Zahlen in das Hexformat umwandeln kann:

#64738 = $FCE2
(RETURN erfolgt nach Eingabe der 8)

Andersherum gelingt es mit dem Befehl '$' und hezadezimaler Angabe. Eine komplette Befehlsübersicht kann durch Eingabe von '?' erzwungen werden.

Anzeige und Modifikation der m-Blocks geschieht durch den Memory-Befehl 'M'. Die Syntax dieses Befehles ist ein wenig aufwendiger als die anderer Anweisungen. Durch Bereichsangabe in der Form von M6-8 oder M4 oder M-5 oder M8- kann der Disc-Monitor mehrere m-Blocks gleichzeitig ausgeben. Mit Parametern in der Art von M4 40 80 kann man die Anzeige eines m-Blocks einschränken auf die Bytes 40 bis 80 (hexadezimal!). Alle Ausgaben des Monitors können durch RUN/STOP unterbrochen und mit SHIFT angehalten werden.

Probieren Sie die einzelnen Befehle am besten einmal aus, und zwar an einer neuen Diskette. Formatiert werden kann sie aus dem DiscMon heraus, durch den Diskettenbefehl "@", der die Sequenz "open 15,18,15," erspart.

In unserem Beispiel also

@ "n:testdiskette,xy" (Die Anführungszeichen sind wahlfrei.)

Über den "Klammeraffen" kann nach dem Formatieren auch das Directory angesehen werden:

@ $

Lesen Sie jetzt den ersten Directory-Block in den Arbeitsspeicher Null ein:

RO 12 00

und sehen sich die Bytes 144 bis 163 (dez.) an:

MO 90 A3

Sie sehen den beim Formatieren gewählten Diskettennamen (Bytes $90 bis $9F) und die Diskettenidentifikation (ID, Bytes $A2/$A3). Diese können jetzt problemlos geändert werden. Entweder schreiben Sie die entsprechenden ASCII-Werte direkt in das Hex-Dump (mit RETURN abschließen), oder Sie benutzen den komfortablen "Text"-Befehl. Sollen die Werte geändert werden in "DISKETTEN-TESTS" als Name und "TD" als neue ID, so geben Sie ein:

TO 90 "DISKETTEN-TESTS " AO AO "TD" (Der Name muß 16 Zeichen lang sein!)

und sehen sich durch Wiederholung des obigen m-Befehls das Ergebnis an. Natürlich muß der so veränderte Block auch auf die Diskette zurückgeschrieben werden, durch

WO 12 00

oder, falls Sie Track- und Sektor-Nummer vergessen haben, durch

BO

Der b-Befehl schreibt immer den entsprechenden m-block an seinen Ursprungsort zurück. Wenn Sie sich das so veränderte Directory ansehen wollen, muß die Floppy vorher neu initialisiert werden (@I), da sonst das DOS (Disc Operating System) weiterhin das alte Directory anzeigt.

Bevor Sie sich den nächsten Directory-Block ansehen, sollten Sie ein kleines Programm auf der Diskette speichern. Verlassen Sie mit "x" und RETURN den Monitor in Richtung BASIC. Nach dem Abspeichern eines kleinen Programmes auf Ihrer Testdiskette (es reicht ein einzeiliges BASIC-Programm) kann der DiscMon mit SYS 49152 neu aufgerufen werden.

Werfen Sie einen Blick auf die ersten beiden Bytes des ja noch in m-Block 0 vorhandenen Disketten-Blocks 12/00 (MO 00 01). Diese zeigen auf Spur 18, Sektor 01. Um die Informationen über den ersten Directory-Block nicht zu verlieren, lesen wir den nächsten Block in m-Block 1 ein:

R1 12 01

Noch einfacher wird diese Operation durch den n-Befehl (next block-Befehl). NO liest den nächsten Block ein (solange bis das Ende-

Kennzeichen 00 FF erreicht ist, siehe oben), soll der nächste m-Block beschrieben werden, so ist die Syntax

N01

Das heißt, der logisch dem in m-Block Null abgelegten Disketten-Block folgende soll in m-Block 1 abgelegt werden.

Die Ausgabe der ersten 40 Bytes dieses m-Blocks durch den PRINT-Befehl (P0 1 00 28) sieht dann folgendermaßen aus:

Hardcopy Track 12/Sektor 01

```
:1 00   00 FF 82 11 00 54 45 53  □[illegible]TES
:1 08   54 50 52 4F 47 52 41 4D  TPROGRAM
:1 10   4D A0 A0 A0 A0 00 00 00  M    □□□
:1 18   00 00 00 00 00 00 03 00  □□□□□□■□
:1 20   00 00 00 00 00 00 00 00  □□□□□□□□
:1 28   00 00 00 00 00 00 00 00  □□□□□□□□
```

Die ASCII-Darstellung rechts im Bild stellt die Steuerzeichen nach der Commodore-Norm dar.

Der Eintrag des ersten Files beginnt bei Byte 2. In den Bytes 3 bis 18 dieses Eintrags steht der Name des abgespeicherten Programms (Leerstellen sind mit Shift-Space, also $A0, aufgefüllt), in den Bytes 0 und 1 sind Track- und Sektornummer des ersten Blocks des Files abgelegt. Der Eintrag des nächsten Files begänne bei Byte 34 dieses Blocks. Diesen Block könnte man jetzt in den Rechnerspeicher laden und, analog der oben für den ersten Directory-Blocks beschriebenen Verfahrensweise, ändern, zurückschreiben und so fort.

Dies soll erst einmal an Grundinformationen zur Diskettenorganisation, reichen. Wollen Sie tiefer in dies Metier einsteigen, sei auf das relativ gute Handbuch zur Diskettenstation verwiesen sowie auf die entsprechende Fachliteratur. Zwei Bücher zu diesem Thema sind am Schluß des Artikels vermerkt.

Vor der vollständigen Befehlstabelle noch eine kurze Erläuterung zu zwei hilfreichen Optionen des Diskmonitors:

Druckerausgabe des M-Befehls wird durch Voranstellen von 'P' erzwungen. Ein weiterer Parameter bestimmt die Formatbreite:
P0 - 40 Zeichen (Bildschirmformat, Plotterbreite 1520)
P1 - 80 Zeichen (Drucker, DIN A4 Ausnutzung)

Mit D kann man sich die Einstellungen der Devicenummern für Drucker und Laufwerk anschauen. Modifiziert wird durch 'DP' und 'DD'.

BEFEHLS-SYNTAX

(mb) = mblock, Ziffer 0-9
(Sc) = Sector
(Sa) = Sekundäradresse
(num)= Hexadezimale Zahl
(Ex) = String o. Hexzahl
(mbr)= Bereich von mblocks

(Tr) = Track
(Dv) = Devicenummer
(Asc)= Stringangabe
(Dz) = Dezimale Zahl
(Pr) = Spezieller Parameter

?	Anzeige aller Befehle in Kurzform
R(mb)(Tr)(Sc)	Block von Diskette lesen. Beispiel: R5 11 03 - liest Track 17, Sektor 3 in m-Block 5
N(mb1)(mb2)	Lädt nächsten Block in mb2 aufgrund der Verkettungszeiger in mb1. Beispiel: N12 - liest dem in m-Block 1 abgelegten logisch folgenden Diskettenblock nach m-Block 2
W(mb)(Tr)(Sc)	m-Block auf Diskette schreiben. Beispiel: W3 01 00 - schreibt m-Block 3 auf Track 1, Sektor 0.
S(mb)	Anzeige von Quelltrack und -sector. Beispiel: S3 - zeigt die Herkunft von m-Block 3 an. Unbenutzte m-Blocks führen zur Meldung "not used".
B(mb)	Schreibt Block auf Diskette zurück. Dabei werden die Quelldaten als Zieltrack und -sector verwendet. Beispiel: B9 - schreibt m-Block 9 auf den Track und Sektor der Diskette, von dem er gelesen wurde.
M ; M(mbr) ; M(mb)(num)(num)	Anzeige der Inhalte der m-Blocks. Beispiel: M3-4 - zeigt den Inhalt der m-Blöcke 3 und 4
P(Pr) ; P(Pr)(mbr) ; ...	Wie oben, jedoch Ausgabe auf Drucker. Der Parameter (0 oder 1) bestimmt die Formatbreite : 40 oder 80 Zeichen. Beispiel: P1 0 00 10 - gibt 80 Zeichen breit den Inhalt des m-Blocks 0 von Byte 0 bis Byte 10 aus.

D	Anzeige der eingestellten Busparameter.
D(Pr)(num)(num)	Ändern der Busparameter für Drucker (Pr="P") oder Laufwerk (Pr="D") Beispiel: DD 09 - ändert die Laufwerknummer der Floppy in 9.
X	Rücksprung zu Basic (Über $0300)
@ ; @(Asc)	Anzeige des Diskettenstatus oder Anzeige einer Directory oder Senden eines Diskettenbefehls. Beispiel: @ S:PROG1 - scratched PROG1
C(mb1)(mb1)	Vergleicht mb1 mit mb2 und gibt differierende Adressen aus.
F(mb)(Ex)	Sucht nach einem beliebigen Ausdruck in m-Block mb. Der Ausdruck kann eine beliebige Mischung von Strings und Hex-Zahlen sein. Beipiel: F0 "text" A0 B4 "wort"
T(mb)(mbr)(Ex)	Schreibt Text (Ex) in m-Block (mb) ab Byte (mbr). Beliebige Mischung von Hex-Zahlen und Text möglich. Beispiel siehe oben.
$(num)	Hexadezimal-Umwandlung.
#(Dz)	Dezimal-Umwandlung.

Technische Hinweise

Der DiscMon kann wie üblich durch CTRL und s aus dem Magazin heraus abgespeichert und nach dem Laden vom eigenen Datenträger mit RUN gestartet werden. Belegt wird der von BASIC nicht benutzte Bereich von $C000 bis $C8B5 (dez. 49152 bis 51381), für die m-Blocks wird der Speicher unter dem BASIC-ROM von $B600 bis $BFFF (dez. 46592 bis 49151) benutzt.
Aufruf/Neustart erfolgt durch SYS 49152. Die Information der m-Blocks bleibt auch nach dem Ausstieg durch "x" erhalten.

Literatur

Schramm, Karsten — Die Floppy 1541, Markt & Technik Verlag, München 1985, 49.- DM

Englisch/Szczepanowski — Das große Floppy-Buch, Data-Becker, Düsseldorf 1983, 49.- DM

COLOR-CODE

Bekannt ist dieses Spiel unter den Namen MASTERMIND oder SUPERHIRN.

Für diejenigen, die das Spiel nicht kennen, hier eine Beschreibung dieser Version. Denn das Programm enthält nur eine kurze Anleitung.

Das Programm erzeugt einen fünfstelligen Farbcode, der sich hinter den Fragezeichen in der ersten Zeile verbirgt. Ihre Aufgabe ist es, diesen Geheimcode herauszufinden. Dieser Farbcode enthält also fünf von sieben möglichen Farben in einer bestimmten Zusammenstellung.

Nachdem Sie eine bestimmte Kombination eingegeben haben, vergleicht das Programm Ihre Eingabe mit den eigenen verschlüsselten Farben. Dabei wird Ihnen mittels des 'X' - Zeichens angezeigt, welche Farben richtig sind.

Dabei gelten folgende Regeln:

Ein weißes X zeigt an, daß eine Farbe richtig ist. Ein schwarzes X zeigt an, daß eine Farbe richtig ist und an der richtigen Position steht. Ein waagerechter Strich bedeutet, daß weder Farben noch Position stimmen. Nachdem der Rechner Ihnen diese Hilfen gegeben hat, liegt es nun an Ihnen, durch richtiges Kombinieren und logisches Denken mit möglichst wenigen Versuchen die richtigen Farben an die richtigen Positionen zu setzen. Der folgende Geheimtip darf nicht gelesen werden: mit der Taste 'H' ist es möglich, sich nach 2, 4, 6 und 8 Versuchen die Positionen 1, 2, 3 und 4 des zu findenden Codes anzeigen zu lassen.

Hier darf wieder weitergelesen werden.
In der Spielart 1 ist jede im Code benutzte Farbe nur einmal vorhanden. In der Spielart 2 kann jede Farbe mehrmals auftreten.

Die Farben können mit den angegebenen Tasten (S = Schwarz, B = Blau und so weiter) ausgewählt werden. Nach dem Setzen der Farbe geht der Pfeilcursor automatisch ein Kästchen weiter. Der Pfeilcursor kann auch mit den Cursor-Tasten für links/rechts bedient werden. Die eingegebene Farbkombination kann mit der RETURN-Taste bestätigt werden.

Na dann knobeln Sie mal....

Funktionen definieren, die funktionieren!

Eine Möglichkeit des C64'er BASIC, die seltener benutzt wird: DEFFN. Mit dieser Anwendung lassen sich pfiffige Tricks entwickeln und Programm-Wiederholungen vermeiden.

DEFFN verarbeitet nur numerische Werte, also nur Zahlen, keine Strings. Trotzdem können auch String-Probleme mit dieser Definition gelöst werden:

Als Beispiel PRINT USINC.

Bei Programmen die Zahlentabellen erzeugen (Abrechnungen, Statistiken oder Ähnliches) ist es oft wünschenswert, die Zahlen so darzustellen, daß die Dezimalpunkte untereinanderstehen.

```
So                         und nicht so

  12.75 DM                 12.75 DM
 123.40 DM                 123.40 DM
    .50 DM                 .50 DM
1024    DM                 1024 DM
```

Das C64'er BASIC kennt hierzu keinen Standard-Befehl.

Natürlich lassen sich Unterprogramme schreiben, die mit Hilfe der üblichen String-Befehle (LEFT$, STR$, MID$, VAL, LEN) das Gewünschte auf den Bildschirm zaubern. Aber über DEFFN wird's eleganter. Keine GOSUBs mit Parameterübergabe, die bei der nächsten Erweiterung nach neuen Zeilennummern verlangen.

Doch wie erkennen, an welcher Stelle der Dezimalpunkt in einer Zahl steht oder stehen sollte?

Verzichten wir zuersteinmal auf den Nachkomma-Teil:

INT (Zahl).

Jetzt gilt es, die PRINT-Position zu finden, ab der die Zahl auf dem Bildschirm erscheinen soll. Da Zahlen im üblichen Dezimalsystem verarbeitet werden sollen, läßt sich das Problem über die mathematische Funktion LOG10 lösen.

LOG10 bedeutet Logarithmus zur Basis 10. Diese Funktion stellt die Umkehrung der Funktion Y=10↑X dar:

Y=LOG10(X)

Jede beliebige Zahl unseres Zahlensystems läßt sich als Exponent von 10 darstellen (Mit Ausnahme der '0'). Der ganzzahlige Anteil liefert dabei zugleich Information über die Anzahl der Stellen:

```
  1 =  10↑0        0 = LOG10(1)
 10 =  10↑1        1 = LOG10(10)
100 =  10↑2        2 = LOG10(100)
... und so weiter
```

Das BASIC verfügt aber nur über die LOG-Funktion, die den natürlichen Logarithmus liefert. Die Lösung ist sogar im C64 Handbuch beschrieben (S.126):

LOG(INT(Zahl))/LOG(10).

Doch wie ist das mit INT(Zahl)=0? Hier wird es kritisch. LOG(0) führt unweigerlich zu einer Fehlermeldung. Schließlich gibt es keine Zahl, die als Exponent eine '0' erzeugen kann:

```
2↑0   = 1
2↑1   = 2
2↑0.5 = 1.41421356.. (=SQR(2))
2↑-1  = 0.5
20↑0  = 1
```

Die Abhilfe:

PU=LOG(INT(Zahl)+.1)/LOG(10)

PU enthält die Anzahl der Stellen vor dem Dezimalpunkt minus 1. Da wir nur den Festkommaanteil von PU zum Positionieren benötigen:

PU=INT(LOG(INT(Zahl)+.1)/LOG(10))

Für Zahl=0 erfolgt keine Fehlermeldung mehr, jedoch wird '0' nicht als Ziffer behandelt. Der Wert für PU ist der gleiche wie zum Beispiel für Zahl=.235. Wir müssen für '0' mit einer IF-Abfrage die Berechnung umgehen.

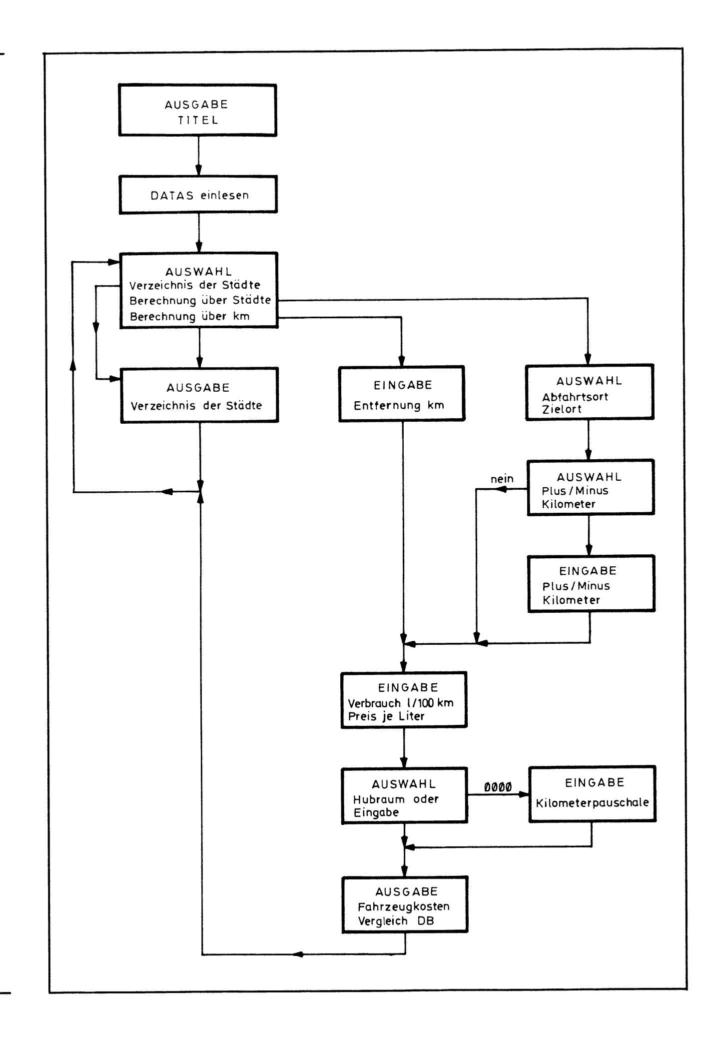
Über TAB(Spalte) können wir unsere Zahlenkolonnen an eine beliebige Stelle auf den Bildschirm setzen.

PRINT TAB(Spalte-PU);Zahl

Sie sehen hier, welche Gedankengänge sich hinter der im ersten Moment verwirrenden Formel verbergen. Vielleicht ein Anreiz Ähnliches einmal selbst zu probieren? Es gibt mit Sicherheit auch völlig andere ebenso elegante Lösungen.

Fortsetzung Listing Reisekosten

```
257 ifz3>63thenz3=1
258 goto250
259 ifz2=z3then237
260 ak=0:q5=0:q3=z3
261 ifz3<48then267
262 q5=z3-47:z9=5:q3=z3
263 syspr,12,5,"Ueber:"
264 syspr,12,20,zl$(q5)
265 z3=iz(q5)
266 qk=ak(q5)
267 z4=abs(z3-z2)
268 z6=z2:ifz2>z3thenz6=z3
269 km=k1(z6,z4)+qk
270 syspr,15,3,"Die Entfernung betraegt ";km;" km"
271 :
272 syspr,20,3,"Eingabe von Plus oder Minus km?"
273 syspr,22,3,"J Ja   N Nein   M Menue":poke198,0
274 getx$:ifx$=""then274
275 ifx$="n"then99
276 ifx$="j"then279
277 ifx$="m"then39
278 goto274
279 syspr,22,3,"P Plus  M Minus  K Keine        ":poke198,0
280 getx$:ifx$=""then280
281 ifx$="k"then99
282 ifx$="p"then285
283 ifx$="m"then288
284 goto280
285 syspr,17,3,"Plus Kilometer:  +       km ":poke198,0:z9=3
286 sysik,17,22,5,x1$,k1$:k1=val(k1$)
287 goto291
288 syspr,17,3,"Minus Kilometer: -       km ":poke198,0:z9=3
289 sysik,17,22,5,x1$,k1$:k1=-val(k1$)
290 ifabs(k1)>kmthen288
291 km=km+k1:goto292
292 syspr,15,1,bl$
293 syspr,15,3,"Die Entfernung betraegt ";km;" km"
294 syspr,20,1,bl$:syspr,22,1,bl$
295 syspr,22,13,"<space> weiter   ":poke198,0
296 getx$:ifx$=""then296
297 ifasc(x$)=32then99
298 goto296
299 print"":print""
300 print"Sie haben einen falschen Ort oder"
301 print"etwas Falsches eingegeben."
302 print"Geben Sie richtig ein oder benutzen Sie"
303 print"die km Direkteingabe!"
304 print"Sie gaben ein:"
305 printz2$:print""
306 printz3$
307 syspr,21,10," w weiter"
308 getx$:ifx$=""then308
309 ifx$="w"then39
310 goto308
311 print"":printy1$
312 return
ready.
```



BEI LADEPROBLEMEN:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft.

Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben.

Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

TONKOPF SELBST JUSTIEREN

Wir haben ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/Wiedergabekopf justieren können.

Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kassettenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage.

Legen Sie nun die zweite Seite von INPUT-64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung "PRESS PLAY ON TAPE" kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar - bei nicht total verstellter Spurlage - völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber

Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken (am besten markieren), und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf!

Der Tonkopf Ihres Recorders ist jetzt exakt justiert. Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie dazu bitte die defekte Kassette mit einem entsprechenen Vermerk an den Verlag ein (Adresse siehe Impressum).

Besitzer der Ausgabe 6/85 können es sich einfacher machen. Das dort veröffentlichte Programm RECORDER-JUSTAGE macht die Einstellung des Daten-Recorders zum Kinderspiel.

Listing justage

```
800 fori=49199to49410:readd:ps=ps+d:pokei,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys12*16↑3+11*16+10
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

Ab 16. September an Ihren Kiosk: INPUT 64 Ausgabe 9/85

Wir bringen unter anderen:

— **MANAGER**

In den Fußspuren von Uli Höneß und Günter Netzer: Werden Sie Manager eines Bundesliga-Vereins Ihrer Wahl. Sie erleben am 64er eine perfekte Simulation des Zusammenspiels von Geschäft und Sport im Kampf um die Tabellenspitze. Die vorgegebenen Daten entsprechen dem aktuellen Stand der Saison 85/86.

— **ReASS**

"Wenn man den Quell-Text zu diesem Maschinensprache-Programm hätte ... ". Dieser Stoßseufzer wird allen geläufig sein, die schon einmal versucht haben, in ein fremdes Maschinenprogramm auch nur kleine Änderungen einzubauen. Dieser Reassembler wandelt auch die undurchsichtigsten Mischungen von Programm-Code und Tabellen zurück in editierbaren Source-Code.

— **und außerdem:**

Serien: Mathe mit Nico, 64er-Tips, SID-Kurs * News * Spiele * Hilfsprogramme * und so weiter.

---

c't-Magazin für Computertechnik

c't 9/85 - jetzt am Kiosk

Prüfstand: Atari 520 ST, Genie 16 C, Schneider CPC 664 * Grundlagen: Störungen aus dem Stromnetz * Projekte: Kompaktes Computernetzteil, SuperTape für CPC 464, Dr. Osborne Kit * Software-Know-how: 65C02-Simulation auf Standard-6502, Schnelle Datensuche durch Hashing, Wie funktionieren Compiler, Newtons Interpolynome auf dem C64 u.v.a.m.

elrad-Magazin für Elektronik

Heft 9/85 - ab 26.8. am Kiosk

* Report: Kleinmischpulte bis 300 D-Mark * Grundlagen: LED-Steuerschaltungen * Bauanleitungen: Subwoofer - Metalldetektor - Fernschalten über Netz - Mod. Vorverstärker (3) - Roadrunner (mobile Kleinst-PA) * u.v.a.m

(Änderungen vorbehalten)

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 1
Tel.: (05 11) 53 52-0

Technische Anfragen
nur dienstags von 9-16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 2746
3000 Hannover 1
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Köhler & Lippmann, Braunschweig.

Druck:
Leunisman GmbH, Hannover
Hahn-Druckerei, Hannover

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.
Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198,-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

ISSN 0177-3771

Titelidee: **INPUT 64**
Titelfoto: Artreferenz

Titelmusik: **INPUT 64**

Programmierung: H. Gehrmann

hier abtrennen

INPUT 64-Abonnement Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat

____

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.
○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.
(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift ______________________________
Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Heise



Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT 64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1